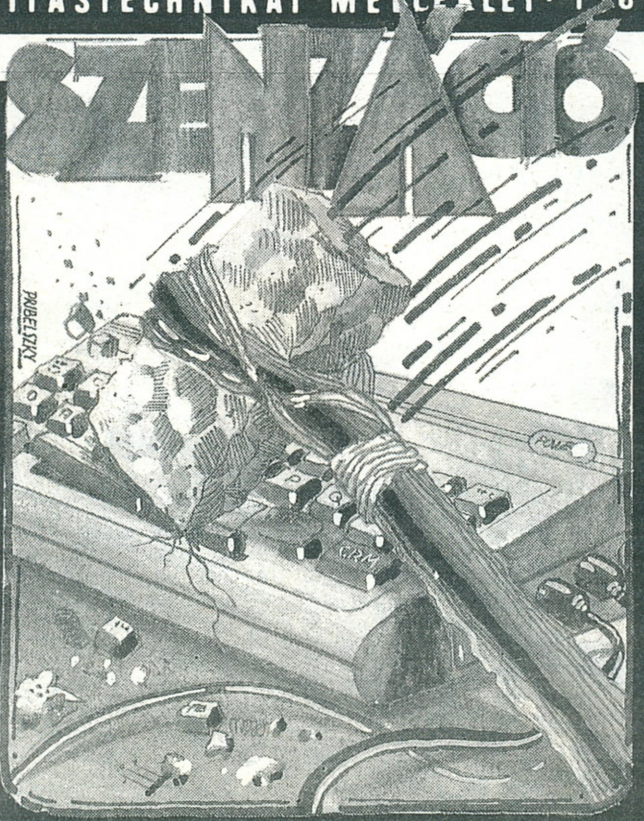


Szenzációs hírt kell megosztanom az olvasóval: barátom, Feri vásárolt egy Commodore 64-es számítógépet. Ahogy az lenni szokott, a gép a család szemefényévé vált. Feri már otthon is el tudja végezni a tervezési munkájához szükséges számításokat, és gmk-jának pénzügyeit is géppel tartja nyilván. Felesége is osztozott a kezdeti lelkesedésben: egy lemez tartalmazza telefonos noteszének adatait, így a géppel keresteti meg népes barátnői körének telefonszámait, és kozmetikusának, fodrásznőjének fogadóóráit. Némi nagyképűséggel szövegszerkesztővel írja leveleit a vidéki rokonoknak – hadd lássa mindenki, hogy milyen „csúcstechnikával" rendelkeznek. A gyerekek, persze nem tudnak betelni a játékprogramokkal. Ők is beléptek általános iskolás osztálytársaiknak abba a kasztjába, akik nap mint nap lázas programcserébe bonyolódnak. Elég gyorsan elsajátították a BASIC alapjait is.

„Mindez szép és jó, de miért volna ez szenzáció – kérdezheti az Olvasó –, hiszen ma már rengeteg háztartásban és irodában megtalálhatók a mikrogépek?"

Mostanáig én sem hittem, hogy egy ilyen hír újságba kívánkozik. A Vasárnapi Hírek március 15-i számának 9. oldalán megjelent rövid hír azonban meggyőzött arról, hogy tévedtem: „COMMODORE. A tagnyilvántartás gépi feldolgozásának előkészítését kezdi meg idén a Szentes és Vidéke Áfész. Egy Commodore 64 típusú számítógépbe körzetek szerint táplálják majd be a felújított nyilvántartást."

Ha ez közérdeklődésre tart számot, akkor nekem is be kellett számolnom barátom háztartásának technikai ugrásáról.

A BIT-LET Olvasói nyilván tudják, hogy a C-64 otthoni játékgépnek készült, és nem igazán alkalmas adatnyilvántartási és -kezelési feladatok ellátására. Vádolhatjuk itt hozzá nem értéssel a Vasárnapi Hírek szerkesztőjét, (meg is tesszük), aki – esetleg más, érdekesebb tudósítás hiányában – közölte a szerinte, a dologhoz fabatkányit sem értő szerint érdekes hírt. Igaz, az újságírónak nem feltétlenül kell értenie a számítástechnikához, de nem is kell(ene) olyan hírt fogalmaznia, amelynek témája olyan számára, akár a szanszkrit. De a legijesztőbbnek mégis azt tartjuk, hogy még mindig akadnak az országban áfészek, béfészek (bocsánat a szó félrehallhatóságáért – szándékos), akik C-64-est vesznek ilyen célra. S mégcsak nem is röstellik a dolgot világgá kürtölni!

Elképzelem az áfész elnökét, amint feltárcsázza újságíró barátját, és izgalomtól elfúló hangon a fülébe kiabálja: „Géza! Hatalmas szenzáció! Gépesítünk!"

Mire az újságíró: „Hogy érted ezt, Bélám? Új fűnyírót vettetek?" Az elnök felháborodva: „Hol élsz, kérlek? Mert mi a XX. században! Számítógépet vásároltunk, Commodore 64 típusút. Nemsokára ez fogja nyilvántartani tagjaink adatait!"

Géza válasz nélkül lecsapja a telefont. Rohan a telexgéphez, hogy ő lehessen az első, aki tudatja a világgal a rendkívüli eseményt. „Számítógép! – gondolja közben – Hiszen erről írnak mostanában mindenhol! Végre én is megfogtam az Isten lábát egy világszenzációval!"

A telexgépek azután a Föld minden táján lekopogják a hírt a csodálatos magyar újdonságról. Az esemény híre eljut Dél-Amerikától Új-Zélandig mindenhová. A világ nagy hírügynökségei az újságok első oldalán, a TV-híradók adásuk első perceiben közlik a szenzációt. Elvégre emögött lenni kell valaminek – mondják –, tudunk mi a sorok között olvasni.

Ahogy az UPI tudósítója fogalmaz: „A fiamnak is volt egy C-64-ese, de kinőtte, így kicseréltük egy nagyobb gépre. A táskámban pedig itt lapul egy 1 MByte-os szövegszerkesztő rendszer, amin a tudósításaimat írom. Hogy a magyarok nem hülyék, az már kiderült a bűvös kockánál. Itt valami hasonló világrengető dologról van szó, ami lehet, hogy sokkal veszélyesebb, mint egy logikai játékszer. De hogy miről?... Én mindenesetre tartok tőlük."

Tallér József

BELÜLRŐL

HÍROLDAL



JÉGTÁBLÁN

Az északi szélesség nyolcvanadik fokán túl, a Wrangel-szigettől több mint ezer kilométerre egy jégtáblán úszik az Északi sark 28 elnevezésű szovjet kutatóállomás. Ilyen kutatóálloMáson most első ízben helyeztek el komplett számítástechnikai, adatgyűjtő és adatfeldolgozó, valamint űrtávközlő rendszert. A komplett rendszer segítségével kibővíthetők a légköri, jég- és óceánmegfigyelési lehetőségek, gyorsabbá és gyakoribbá válik a fontos információk eljuttatása a meteorológiai központokhoz és az Északi-tengeri úton haladó hajók kapitányaihoz.

SZELLEMKÚT

Sikertelennek tűnő kezdeményezésre került sor a szomszédos Ausztriában. Teljesen személyzet nélküli, „szellembenzinkutakat'' állítottak üzembe. A kút vezérlőegységén be kellett táplálni a kívánt típusú és mennyiségű benzin adatait, majd bedobni a gép által meghatározott papírpénzt. A kút számítógépe megvizsgálta a pénzt, és ha mindent rendben talált, kiadta a szükséges benzinadagot. A megoldás előnye, hogy nem kell személyzet és a kút éjjel-nappal üzemelhet. A „szellemkutas'' vállalkozás azonban nem vezetett sikerre. Először a konkurencia támadt: szintén éjjel-nappal üzemeltette a kezelőszemélyzetes kutait. Majd a hatóság kegyelemdöfése következett: egy régi rendelkezésre hivatkozva — mely szerint erősen tűzveszélyes anyagokat felügyelet nélkül hagyni tilos — elrendelte, hogy legyen a kútnál felügyelő személy. Így viszont megszűnt az automatizálás bérmegtakarítást eredményező előnye és ezzel együtt az új megoldás értelme is.

TÁROLÓS CSATLAKOZÓK

A személyi számítógép és a nyomtató közötti puffertár praktikus elhelyezését oldotta meg a svájci Wiesemann cég. A 64, illetve 256 Kbyte kapacitású tárakat a csatlakozódugaszban helyezték el, így azok külön helyet nem igényelnek. A tárak különböző, egyéb funkciók ellátására is programozhatók. A programok kikapcsoláskor nem törlődnek, viszont átírhatók.

KÖZÖS KÁRTYA

Brüsszelben a Közös Piac Bizottsága egy olyan közös, számítógépes kódrendszer bevezetését javasolta, melynek segítségével az egyes tagországokban forgalomban levő hitelkártyákat bármelyik másik tagországban akadálytalanul használni lehetne. A kódrendszer és a szükséges új banki berendezések kidolgozása és üzembe helyezése összesen nyolcszázhatvanmillió dollárba kerülne.

GYORSSZÁMOLÓ

Az Egyesült Államok-beli Thinking Machine cég újdonságaként egy különleges számítógép kifejlesztéséről kaptunk hírt. A párhuzamos processzorral dolgozó gép másodpercenként több mint egybillió műveletet végez. Az új gyorsszámoló teljesítményére jellemző, hogy hatvannégy ezer számot harminckét mikromásodperc alatt ad össze. A speciális számítógép jól használható például VLSI áramkörök szimulációjára és kifejlesztésére.

-CLN

ZALAI PAPRIKÁS

Bővítette és korszerűsítette zalaegerszegi feldolgozóüzemét a Zala Megyei Állatforgalmi és Húsipari Vállalat. Berendezkedtek a hosszú szavatossági idejű szalámik gyártására is. Az ehhez szükséges üzem már a múlt év végén felépült. Az itt kialakított, klimatizált gyorsérlelő kamrákban számítógépvezérléssel készül a zalai paprikás és több kolbászféleség. Ezekből összesen négyszáz tonnányit állítanak elő ebben az évben.

VIDEOMÁTRIX

A fenti néven különleges elektronikus eredményjelző táblát gyárt exportra az Elektronikai Gyár. A szabadalmaztatott videomátrix-rendszer lehetővé teszi, hogy az eredményjelző táblán egyszerre akár külön-külön szöveges és képi információ jelenjék meg: a számítógép és a video összekapcsolása révén a tábla közel százezernyi fénypontján — mint valami hatalmas televíziós képernyőn — nyomon követhetők a stadionban zajló események, egyes jelenetek tetszés szerint bármely színárnyalatban visszajátszhatók, lassíthatók.

AKUPUNKTÚRA

A világ egyre több országában egyre nagyobb számban veszik igénybe különféle betegségek gyógyítására a tűszúrásos gyógymódot, az akupunktúrát. A régi keleti orvostudomány több szocialista országban is polgárjogot nyert. A moszkvai fizikoterápiás kutatóintézetben például akupunktúrával gyógyítják az asztmát, a neurózist, a vegetatív keringési zavarokat, a mozgásszervi megbetegedéseket. A moszkvai orvosok azonban úgy vélik, hogy ezzel távolról sem merült ki az akupunktúra alkalmazási lehetősége. Hogy újabb betegségek gyógyítására felhasználható legyen, számítógépes kutatómunkát végeznek. Ismeretes az, hogy a legtöbb betegség nem egy, hanem több akupunktúrás testpont ingerlése révén gyógyítható. Így a számítógép feladata, hogy megrajzolja a beteg akupunktúrás portréját és meghatározza, mely pontokat milyen intenzitással kell stimulálni.

SZALAGÉGETÉS

Az adatvédelemmel szembeni elvárások egyre inkább megkövetelik a már nem használt mágnesszalagok megsemmisítését. Az Egyesült Államokban azokat a mágnesszalagokat, amelyek fontos adatokat tartalmaznak, biztonsági okokból csak teljes törlés után szabad megsemmisíteni. A BASF nyugatnémet cég pedig a teljes megsemmisítésre az elégetést ajánlja, melynek során nem várható káros anyagok keletkezése. Az NSZK-ban ma már léteznek olyan vállalkozók, amelyek kulcsra zárható szállítótartályokban viszik el a szalagokat és a megrendelő cég adatvédelmi megbízottja jelenlétében végzik el az égetést.

BRAILAB

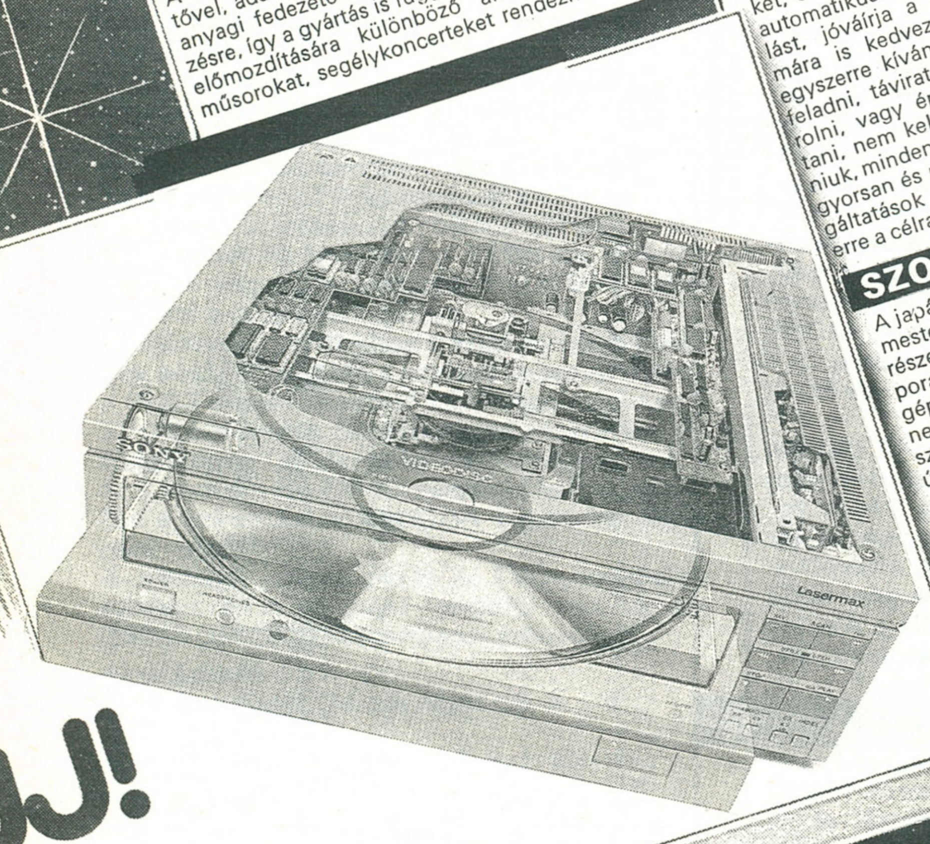
Vakok számára fejlesztette ki Brailab elnevezésű számítógépét másokkal együtt a KFKI egyik munkatársa és programozóként dolgozó vak felesége. Az Arató András által konstruált masina legfőbb sajátossága, hogy a beütött adatokat, programokat szintetizált hangon, magyarul „beszélve" visszajelzi. Harminc ilyen számítógép készült már el, s további ötven gyártása várható. Ezek közül többet már használnak is tanulási célokra a Vakok Szövetségében és általános iskolájában. A továbbfejlesztett változat szövegszerkesztővel, adatbázissal is kiegészül. A kifejlesztés anyagi fedezete egyelőre nem áll rendelkezésre, így a gyártás is függőben van. A munka előmozdítására különböző akciókat, zenei műsorokat, segélykoncerteket rendeznek.

POSTAGÉPEK

A posta tervezi, hogy a tavaly indított sikeres soproni kísérlet nyomán fokozatosan több postahivatalban is bevezeti a számítógépes kiszolgálás rendszerét. Ennek elősegítésére az idén 20 millió forintért vásárolnak mikroszámítógépeket. Az ügyfelek – a csomagfeladás kivételével – bármely ablaknál valamennyi postai szolgáltatást igénybe vehetik. A számítógép tárolja az egyes szolgáltatások díjtételeit, a különböző tarifákat és más adatokat, így például valutaárfolyamokat, neveket, címeket. Könyveli, összesíti a forgalmat, automatikusan elkészíti a napi elszámolást, jóváírja a kamatot. Az ügyfelek számára is kedvező az új rendszer, mert ha egyszerre kívánnak például levelet és pénzt feladni, táviratozni, képeslapot, lottót vásárolni, vagy éppen takarékbetétkönyvet váltani, nem kell más-más ablaknál sorban állniuk, mindent egy helyen elintézhetnek. A gép gyorsan és megbízhatóan segít a kívánt szolgáltatások teljesítésében. Az évtized végéig erre a célra hatvanmillió forintot fordít a posta.

SZOFTVERÍRÓ

A japán ötödik generációs számítógép, illetve mesterséges intelligencia kutatások egyik részeredményeként készítette el az NC Corporation japán cég automatikus, szoftveríró gépét. A Cobol programnyelven író rendszernek kezelője élőszóval közli az elkészítendő szoftver jellemzőit. A rendszer hozzákezd az új programtermék elkészítéséhez, miközben a felmerülő problémákat a kezelőtől folyamatosan megkérdezi. Az új, különleges automata fontos előnye az is, hogy élőszóval irányítható, így nincs szükség hozzá az amúgy is kevés számúnak bizonyuló programozó táborra.



ÚJ!

PORTÁS

Számítógépes portás működik Szolnok megye legnagyobb gépipari üzemében, a Jászberényi Hűtőgépgyárban. A központi gyár portáin elektronikus rendszer rögzíti és tartja nyilván, hogy ki érkezett, ki távozik. Regisztrálja azt is, hogy a dolgozó főmunkaidőre, túlórára vagy gmk-munkára érkezik a gyárba, illetve a munkaidőből mennyit volt távol. A gyári kapuknál elhelyezett, műanyag kártyával működő készülékek egy számítógéphez csatlakoznak, ez személyre szólóan összegyűjti és feldolgozza az adatokat. A vállalati munkacsoport által kidolgozott rendszer szolgáltatja az alapot a havi bérelszámoláshoz, a dolgozók csak annyi óra után kaphatnak bért, amennyit ténylegesen ledolgoztak.

SONY LDP-2000

Az utóbbi években a különböző elektronikus adattárolási módok összefonódásának lehetünk a tanúi. Úgy tűnik a jövő leghatékonyabb eszközei a video, audio, s a komputertechnika egyes elemeinek összeépítéséből jön létre. A video-lemezjátszók például univerzális adattároló eszközökké válnak. Ennek a Sony LDP–2000 típusú berendezésnek a hirdetéséből például megtudtuk, hogy minden olyan számítógéppel használható, amely RS 232C vagy IEEE–488 paralell interface-val rendelkezik. Ráadásul a kommunikációs buszon keresztül egyidőben 14 felhasználót tud kiszolgálni. A kapacitása sem semmiség, egyetlen lemezoldalon 221 Megabyte információ fér el.

ROBOTDORFER

Új lehetőségeket jelent a zenerajongóknak a híres Bösendorfer cég számítógépes robotzongorája. A robotzongora jelentősége abban áll, hogy rögzíteni, majd automatikusan újrajátszani képes a nagy zongoravirtuózok felejthetetlen játékát. A Bösendorfer cég újítói optikai érzékelőrendszereket helyeztek el a zongora billentyűihez és kalapácsaihoz. Így optikai módon történik a leütött hangok megfelelő ideig és magasságon történő érzékelése, sőt további érzékelőkkel mintegy kétszázötven pedálnyomási szög mérése. Az érzékelt értékeket digitális jelek formájában lemezen vagy szalagkazettán rögzítik. A művész játékát követően ugyanazon zongora felhasználásával és a lemez vagy szalagkazetta segítségével a legbrilliánsabb zenei produkció is teljes hűséggel visszaadható. Ilyenkor az történik, hogy a rögzített digitális jelek mágneseket mozgatnak, amelyek speciális mechanizmuson keresztül működtetik a billentyűket és a pedált. Az új önjátszó zongorák különleges számítógép programokkal kerülnek forgalomba, melyek révén lehetőség van például hibajavításra, a hamisan vagy gyengén leütött hangok korrekciójára, sőt a hangszer hangolására is.

KÖRRAJZOLÓ RUTIN

Az alábbi kis programmal az EXT utasítással rajzolhatunk köröket.

A kör rajzolása az EXT sorszám, x, y, r utasítással történik, ahol x és y a kör középpontja, r pedig a sugara. Mivel az EXT paraméterei a —32768-tól +32767-ig terjedő tartományba eshetnek, ezért a képernyőn kívül eső középpontú kört is rajzolhatunk! A rutin a kört egy 806 oldalú sokszöggel közelíti, így csak azokat a szakaszokat rajzolja meg, melyeknek mindkét végpontja a képernyőn belül van. Természetesen a rajzolás az aktuális tintaszín, MODE és STYLE paraméterek szerint történik, de a STYLE vonaltípus csak nagyobb átmérőknél érvényesül.

Az első lefuttatás után EXT0,512,480,400 paranccsal próbáljuk ki a rutint. Ha nem kaptunk kört, akkor hiba van a DATA sorokban, javítsuk ki! Ha kört kaptunk, akkor a DATA sorok már feleslegesek, kitörölhetők a "DELETE 4–" paranccsal. (A gépi kódú program az első sorokba került.) Az így elkészült program magnóra kimenthető, későbbi programokban felhasználható.

Fontos apróságok:

A rutin csak akkor működőképes, ha a szokásos BASIC program kezdőcímről (6639) LOMEM utasítással nem mozdítjuk el. A programot csak a 3. sortól listázzuk, mert az első sorok listázása a többiét is megzavarhatja.

A körrajzoló algoritmus leírása:

Ha az (x;y) pontot az origó körül szöggel elforgatjuk, új koordinátáit (ux;uy) az

$ux=x*\cos(\varphi)+y*\sin(\varphi)$

$uy=y*\cos(\varphi)-x*\sin(\varphi)$

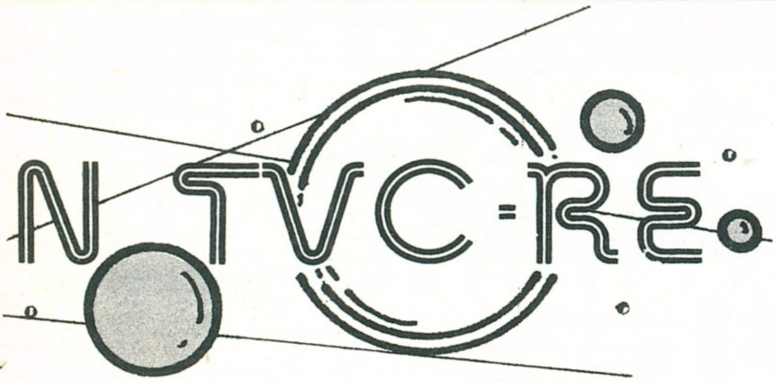
képletek segítségével számíthatjuk ki. Ha az elforgatást többször megismételjük, és a kapott pontokat összekötjük, sokszöget kapunk. Nem origó középpont esetén a középpont koordinátáit (kx,ky) a kapott pontokhoz hozzá kell adni. A rutin a kört egy 806 oldalú sokszöggel közelíti. Ebben az esetben $\sin(\varphi)\approx 1/128$, $\cos(\varphi)\approx 32767/$

A szerkesztő azért van,

hogy a lap olyan legyen,

amilyenek az olvasói!

```
                              ORG 6639
250                           DB 250    ; sorhossz
  0,   0                      DW 0      ; sorszam
254                           DB 254    ; ! tokenje
  0,   0                   KX:DW 0
  0,   0                   KY:DW 0
  0,   0,   0,   0          X:FILL 4,0
  0,   0,   0,   0          Y:FILL 4,0
  0,   0,   0,   0         UX:FILL 4,0
  0,   0,   0,   0         UY:FILL 4,0
  0,   0,   0,   0        VAR:FILL 4,0
  0,   0,   0,   0       VAR2:FILL 4,0
  1,   4,   0            MOVE:LD BC,4
237,176                       LDIR
201                           RET
  6,   4                MINUS:LD B,4
183                           OR A      ; CY:=0
 26                        M1:LD A (DE)
158                           SBC A,(HL)
 18                           LD (DE),A
 35                           INC HL
 19                           INC DE
 16,249                       DJNZ M1
201                           RET
  6,   0               SIN_FI:LD B,0
126                           LD A,(HL)
 23                           RLA
 48,   1                      JR NC,S1
  5                           DEC B     ; B:=255
 43                        S1:DEC HL
 43                           DEC HL
 17,   7, 26                  LD DE,VAR
213                           PUSH DE
229                           PUSH HL
197                           PUSH BC
205,  15, 26                  CALL MOVE
241                           POP AF
 50,  10, 26                  LD (VAR+3),A
225                           POP HL
 43                           DEC HL
126                           LD A,(HL)
 23                           RLA
225                           POP HL
203,  22                      RL (HL)
 35                           INC HL
203,  22                      RL (HL)
 35                           INC HL
203,  22                      RL (HL)
 35                           INC HL
203,  22                      RL (HL)
201                           RET
  1,   0,   0          COS_FI:LD BC,0
126                           LD A,(HL)
 23                           RLA
 48,   1                      JR NC,C1
 11                           DEC BC    ; BC:=65535
 43                        C1:DEC HL
229                           PUSH HL
197                           PUSH BC
 17,  11, 26                  LD DE,VAR2
205,  15, 26                  CALL MOVE
193                           POP BC
237,  67, 13, 26              LD (VAR2+2),BC
225                           POP HL
 43                           DEC HL
229                           PUSH HL
126                           LD A,(HL)
 23                           RLA
 33,  11, 26                  LD HL,VAR2
203,  22                      RL (HL)
 35                           INC HL
203,  22                      RL (HL)
 35                           INC HL
203,  22                      RL (HL)
```



32768. Mindkét értékkel viszonylag könnyen lehet binárisan szorozni. E szorzásokat a SIN_FI és COS_FI szubrutinok végzik az n/128=2∗n/256 és 32767/32768∗n= n−n/32768 számítások alapján. Az ux, uy, x, y változók és var, var2 részeredmények tárolása 32 biten történik. (16 bit egész rész, 16 bit tört rész.) A középpont kx, ky koordinátáinak tárolása 16 biten történik. Az összeadást és kivonást a PLUS és MINUS rutinok végzik, a változókat a MOVE rutin helyezi a megfelelő helyre.

A kiszámolt pontokhoz a középpont értékének hozzáadását, és a szakasz megrajzolását a PLOT címen kezdődő rész végzi. Sajnos a TVC koordináta rendszerének vízszintes és függőleges léptékezése nem azonos, ezért az x koordináta 7/8-ad részével kell rajzolni. Ezt a számítást a ZSUGOR címen kezdődő rész végzi.

A vonalak meghúzását a TVC operációs rendszerének rutinjai végzik, melyek az RST48 (RST30H) utasítással érhetők el.

Bata László,
1137 Budapest, Jászai Mari tér 4. I/4.

```
10 !*******    / osszesen 245 db. csillag
20 !*****      / osszesen  63 db. csillag
30 POKE 33,130: POKE 34,26: REM EXT 0,x,y,r
40 FOR I=6639 TO 6956: READ A: POKE I,A: NEXT
50 DATA
60 DATA
```

(A DATA sorokba az Assembler lista bal oldalán lévő adatokat kell írni!)


```
225                     POP HL
 43                     DEC HL
 17,  7, 26             LD DE,VAR
213                     PUSH DE
205, 15, 26             CALL MOVE
209                     POP DE
 33, 11, 26             LD HL,VAR2
205, 21, 26             CALL MINUS
201                     RET
247,9               VEG:RST 48,9   ; toll fel
                    247,  9    VEG:RST 48,9 ; toll fel
251                     EI
201                     RET        ; a program vege
243             KEZDET:DI          ; a program eleje
 34,243, 25             LD (KX),HL
237, 83,245, 25         LD (KY),DE
237, 67,253, 25         LD (Y+2),BC ; (Y):=r
 33,  0,  0             LD HL,0
 34,251, 25             LD (Y),HL
 34,247, 25             LD (X),HL
 34,249, 25             LD (X+2),HL
 33, 39,  3             LD HL,807; ciklushossz
229                     PUSH HL
225             CIKLUS:POP HL
 43                     DEC HL
 17,  0,  0             LD DE,0
183                     OR A       ; CY:=0
237, 90                 ADC HL,DE; HL=0 ?
 40,214                 JR Z,VEG ; kesz a kor!
229                     PUSH HL
 42,245, 25        PLOT:LD HL,(KY)
237, 91,253, 25         LD DE,(Y+2)
 25                     ADD HL,DE; HL:=y+ky
235                     EX DE,HL ; DE:=y koord.
 42,249, 25      ZSUGOR:LD HL,(X+2)
 68                     LD B,H
 77                     LD C,L
203, 40                 SRA B
203, 25                 RR C
203, 40                 SRA B
203, 25                 RR C
203, 40                 SRA B
203, 25                 kR C       ; BC:=HL/8
183                     OR A       ; CY:=0
237, 66                 SBC HL,BC; HL:=7/8 HL
237, 75,243, 25         LD BC,(KX)
  9                     ADD HL,BC; HL:=x+kx
 68                     LD B,H
 77                     LD C,L     ; BC:=x koord.
247,  6                 RST 48,6 ; pozicionalas
183                     OR A       ; hiba tortent?
 40,  4                 JR Z,PL1 ; nem volt hiba
247,  9                 RST 48,9 ; toll fel
 24,  2                 JR PL2
247,  8             PL1:RST 48,8 ; toll le
 33,250, 25         PL2:LD HL,X+3
205, 71, 26             CALL COS_FI; var:=x*cos(fi)
 33,  7, 26             LD HL,VAR
 17,255, 25             LD DE,UX
213                     PUSH DE
229                     PUSH HL
 24,  5                 JR UJ_SOR+4
255                     DB 255     ; sorveg
 68              UJ_SOR:DB 68      ; sorhossz
  0,  0                 DW 0       ; sorszam
254                     DB 254     ; ! tokenje
205, 15, 26             CALL MOVE; ux:=var
 33,254, 25             LD HL,Y+3
205, 32, 26             CALL SIN_FI; var:=y*sin(fi)
225                     POP HL
209                     POP DE
  6,  4            PLUS:LD B,4
183                     OR A       ; CY:=0
 26                  P1:LD A,(DE)
142                     ADC A,(HL)
 18                     LD (DE),A
 35                     INC HL
 19                     INC DE
 16,249                 DJNZ P1    ; ux:=ux+var
 33,254, 25             LD HL,Y+3
205, 71, 26             CALL COS_FI; var:=y*cos(fi)
 33,  7, 26             LD HL,VAR
 17,  3, 26             LD DE,UY
213                     PUSH DE
229                     PUSH HL
205, 15, 26             CALL MOVE; uy:=var
 33,250, 25             LD HL,X+3
205, 32, 26             CALL SIN_FI; var:=x*sin(fi)
225                     POP HL
209                     POP DE
205, 21, 26             CALL MINUS ; uy:=uy-var
  1,  8,  0             LD BC,8
 33,255, 25             LD HL,UX
 17,247, 25             LD DE.X
237,176                 LDIR       ; x:=ux, y:=uy
195,158, 26             JP CIKLUS
255                     DB 255     ; sorveg
```

ötlet



a GEOS óra átállítása

A GEOS-rendszer számos szolgáltatást nyújt a felhasználónak. Az eddig piacre került két felhasználói program (a GEOPAINT és a GEOWRITE) mellett fontos szerepet játszanak az úgynevezett segédprogramok (DESK ACCESSORIES) is. Segítségükkel pl. munka közben számításokat végezhetünk, jegyzeteket készíthetünk vagy figyelemmel kísérhetjük az idő múlását. Ez utóbbit a PREFERENCE MANEGER, illetve az ALARM CLOCK teszi számunkra lehetővé. Fontos ez a funkció a file-ok nyilvántartásánál is, hiszen minden file esetében letárolja a rendszer az információs szektorban az aktuális időt.

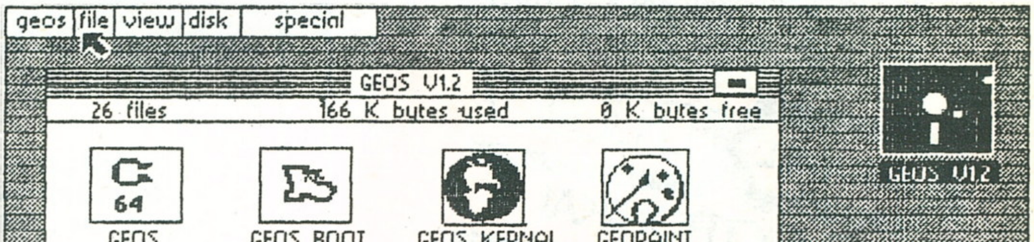
Az európai terjesztésnél a program készítői nem vették figyelembe az amerikai és az európai hálózati frekvencia eltérését (USA 60 Hz, Európa 50 Hz), ami a rendszer órájának jelentős késését okozza. Az ebből eredő kellemetlenséget megszüntethetjük, ha a GEOS KERNAL-ban egy apróbb változtatást hajtunk végre.

Ezt úgy tehetjük meg, hogy elhagyjuk a GEOS-rendszert (SPECIAL parancs, BASIC utasítás) és POKE 53132,128 utasítással átállítjuk a CIA-t 50 Hz-re, majd SYS 52298 utasítással visszatérünk a GEOS-ba. Így elértük, hogy a gép kikapcsolásáig helyesen fogja az óránk mérni az időt.

Ha ez nem elég és nem akarjuk minden egyes indításkor ezt a bosszantó műveletet végrehajtani, akkor írjuk át a lemezen a GEOS KERNAL-t. Ebben segít a mellékelt BASIC program. Ha sikerült a programot helyesen begépelni, tegyük be a meghajtóba a GEOS lemezünk biztonsági másolatát és indítsuk el a programot. Ha eredményesen lefuttattuk a programot, akkor ellenőrizzük le, hogy mennyivel lett pontosabb az óra. Természetesen abszolút pontosságot senki ne várjon egy frekvenciafüggő órától, hiszen nálunk a terheléstől függően 48–51 Hz között ingadozik a hálózati frekvencia.

Honti Tamás

```
10 :
20 REM > GEOS SEGEDPROGRAMOK  1. <
30 REM
40 REM >>>>      EURO-HERZ      <<<<
50 REM
60 REM >> IRTA: HONTI TAMAS '87 <<
70 REM
80 REM >      SZEKESFEHERVAR       <
90 :
100 PRINTCHR$(147)CHR$(144):POKE53280,.:POKE53281,30
110 PRINT" KERESEM A GEOS LEMEZEDEN AZ ORA RUTINT"
120 A$(.)=CHR$(13):A$(1)=CHR$(220)
130 A$(2)=CHR$(3):A$(3)=CHR$(127)
140 A$(4)="GEOS KERNAL"
150 OPEN 15,8,15,"I":GOSUB350
160 OPEN 2,8,2,A$(4):GOSUB350
170 GET#2,A$,A$,T$,S$:CLOSE 2
180 OPEN 8,8,8,"#1"
190 FOR J= 1 TO 17
200 PRINT#15,"U1:"8;.;ASC(T$);ASC(S$+CHR$(0))
210 GET#8,T$,S$:NEXT
220 GET#8,A$:IF A$<>A$(I) THEN I=.
230 I=I+1:IFI<4THEN220
240 PRINT:PRINTSPC(14)"MEGTALATAM !!"
250 PRINT:PRINTSPC(10)"ATIRJAM 50 HZ-RE ? <I/N>":PRINT
260 GETA$:IFA$=""THEN260
270 PRINTSPC(17)A$;:FORI=1TO100:NEXT:IFA$="N"THENPRINT"EM":GOTO 400
280 IFA$="I"THENPRINT"GEN !":GOTO300
290 PRINT:PRINTSPC(4)"]CSAK IGENT VAGY NEMET VALASZOLJ !":GOTO250
300 PRINT#8,CHR$(128);
310 PRINT#15,"M-W"CHR$(1)CHR$(.)CHR$(1)CHR$(144):GOSUB350
320 PRINT:PRINTSPC(11)"AZ ATIRAS KESZ !"
330 GOTO400
350 INPUT#15,A,A$:IFA=.THENRETURN
360 IF A=62THENPRINTSPC(3)"A LEMEZROL HIANYZIK A "A$(4)
370 IFA=74 ORA=21THENPRINTSPC(7)"NINCS LEMEZ A DRIVE-BAN !!"
380 IFA=26THENPRINTSPC(8)"VEDD LE  AZ IRASVEDELMET !!"
390 PRINT" JAVITSD KI A HIBAT ES PROBALKOZZ UJRA !"
400 CLOSE15:CLOSE8:END
```



```
60000 :
60001 ONERRORGOTO60006
60002 IFPEEK(16548)<>2340RPEEK(16549)<>67T
HENPRINT" Az 'átsorszámozó kész!":POKEE165
48,234,67:END
60005 GOTO60180
60006 IFERR<>10THENPRINTERL"sor,"ERR:STOP
60007 IFW>=32768THENW=W-65536ELSEPRINT"*";
ERL:STOP
60008 RESUME
60010 :
60020 FU=C1+4:NR=NR+1
60030 W=FU:PF=PEEK(W)
60040 IFFU=C2-1THENRETURN
60050 IFPF=147THENRETURN
60060 IFPF=0THENRETURN
60070 IFPF=1410RPF=1450RPF=1590RPF=2020RPF
=149THEN60090
60080 FU=FU+1:GOTO60030
60090 Z$="":JE=0
60100 FU=FU+1:W=FU:PF=PEEK(W)
60110 IFPF=32THEN60140
60111 IFPF<58ANDPF>47THENJE=1:GOTO60140
60120 IFLEN(Z$)<5THEN60135
60130 IFPF=44THEN60090ELSE60040
60135 IFJE<1THEN60040ELSEPRINT"Nincs elég
hely a hivatkozott sorszámnak a"SO"sorrban
!":END
60140 Z$=Z$+CHR$(PF):GOTO60100
60150 W=C2+2:SO=PEEK(W):W=C2+3:SO=SO+256*P
EEK(W)
60160 C1=C2:W=C1:C2=PEEK(W):W=C1+1:C2=C2+2
56*PEEK(W)
60170 RETURN
60180 REM kezdet
60190 C2=17386
60200 GOSUB60150:IFC1=0ORSO>=60000THEN6022
0
60210 GOSUB60010:GOTO60200
60220 :
60230 PRINTCHR$(7);:INPUT" Mekkora a növek
mény";NV
60240 IFNV<1THENPRINT"Rossz növekmény!":GO
TO60230
60250 IF(NR-1)*NV>=60000THENPRINT"Túl nagy
 növekmény!":GOTO60230
60260 CLS:KK=0:K0=256*PEEK(20)
60270 C2=17386
60280 GOSUB60150:IFC1=0ORSO>=60000THEN6046
0
60290 GOSUB60300:GOTO60280
60300 FU=C1+4
60310 W=FU:PF=PEEK(W)
60320 IFFU=C2-10RPF=1470RPF=0THENRETURN
60330 IFPF=1410RPF=1450RPF=1590RPF=2020RPF
=149THEN60350
60340 FU=FU+1:GOTO60310
60350 A1=FU:Z$="":JE=0
60360 FU=FU+1:W=FU:PF=PEEK(W):IFPF=32THEN6
0360
60370 IFPF<58ANDPF>47THENZ$=Z$+CHR$(PF):JE
=1:GOTO60360
60380 IFJE<1THEN60320ELSERC=VAL(Z$)
60390 GOSUB60410:W=A1:A2=WAND255:W=RC:R2=W
AND255:W=KF:POKEW,A2,INT(A1/256),R2,INNT(R
C/256):BEEP40,200;50,200;30,800;60,400
60400 KK=KK+1:IFPF=44THEN60350ELSE60320
60410 KF=K0+KK*6:RETURN
60420 KK=0:W=UC:U1=WAND255:U2=INT(UC/256):
W=SO:S1=WAND255:S2=INT(SO/256)
60430 KF=K0+2+KK*6:W=KF:IFPEEK(W)<>0THEN60
440
60431 W=KF+1:IFPEEK(W)<>0THEN60440
60432 W=C1+2:POKEW,U1,U2:BEEP30,3:RETURN
60440 W=KF:IFPEEK(W)<>S1THEN60450
60441 W=KF+1:IFPEEK(W)<>S2THEN60450
60442 W=KF+2:POKEW,U1,U2:BEEP40,4
60450 KK=KK+1:GOTO60430
60460 C2=17386:JE=0
60470 GOSUB60150:IFJE<1THENUC=SO:JE=2ELSEU
C=UC+NV
60480 IFC1=0ORSO>=60000THEN60570
60490 GOSUB60420:GOTO60470
60500 W=A1
60510 W=W+1:PF=PEEK(W):IFPF=320R(PF<58ANDP
F>47)THENPOKEW,32:GOTO60510
60520 U$=STR$(UC):IFLEN(U$)>5THENU$=RIGHT$
(U$,5)
60530 FORJ=1TOLEN(U$)
60540 W=A1+J:POKEW,ASC(MID$(U$,J,1)):BEEP6
0,60;5,100
60550 NEXTJ
60560 RETURN
60570 KK=0
60580 GOSUB60410
60590 W=KF:A1=PEEK(W):W=KF+1:A1=A1+256*PEE
K(W):W=KF+4:UC=PEEK(W):W=KF+5:UC=UC+2556*P
EEK(W)
60600 IFA1=0THEN60620
60601 IFUC=0THEN60615
60610 GOSUB60500
60615 KK=KK+1:GOTO60580
60620 CLS:ONERRORGOTO0:PRINT"Kész az átszá
mozás!"
```


SZOFTVER ÖTLETEK



RENUMBER PRIMO

Az átsorszámozó program használata

A) Ha körültekintőek vagyunk, akkor már programszerkesztéskor úgy írjuk le a GOSUB, GOTO, RESUME, ELSE, THEN utáni sorszámot, annyi szóközzel előtte vagy utána, hogy a helyén elférjen egy másik ötkarakteres szám is. Ha megfeledkezünk róla, az sem baj, a program majd figyelmeztet rá.

B) Az újra sorszámozandó program ne tartalmazzon 60000-nél nagyobb sorszámot!

C) Betöltés. Az átsorszámozandó program a gépben van! Parancsmódban beírjuk:

C=PEEK(16633)+256∗PEEK(16634):C=C–2:C2=INT(C/256):
:C1=C–256∗C2:POKE16548,C1,C2:CLEAR

D) Betöltés után egyszer lefuttatjuk, akkor visszaállítja a BASIC eleje mutatót: RUN60000

E) Ha még egyszer lefuttatjuk a RUN60000 paranccsal (vagy akárhányszor!), az újra sorszámozó program először megvizsgálja a hivatkozásokat, a GOSUB, GOTO, ELSE, THEN és RESUME utáni számokat, hogy módosítás esetén lesz-e elég hely egy ötjegyű számot beírni. Ha valahol nincs elég, a sor számát kiírja és megáll. Ebben az esetben a sorszám mellé beírunk néhány szóközt, „EDIT" üzemmódban, utána a RUN60000-rel ismét elindíthatjuk.

A vizsgálat után megkérdezi a sorszámok lépésközét. Helytelen érték megadása esetén hibajelzést ad. Az átsorszámozandó program kezdő sorának száma megszakad.

Futás közben hangok jelzik, hogy a gép él.

A befejezést a „Kész az átszámozás" felirat jelzi, közben a képernyőt használja adattáblának.

Koszper Vilmos

INPUT RUTIN PRIMO

Programozás közben sokszor szükség van olyan input rutinra, amely „elfogadja" a" ;, és a: karaktereket. A program az inputnál vár, és mást nem tud elvégezni. PRIMO-ra szeretnék egy INPUT RUTIN-t közreadni, ami megoldja ezeket a problémákat.

A tulajdonképpeni rutin az 1000. sorban kezdődik.

Az alkalmazás megértéséhez 5–80. sorok adhatnak magyarázatot. A 1020. sorba megfelelően elhelyezett GOSUB-bal például kiírathatjuk a kérdésre használt gondolkodási időt folyamatosan. A programrészletet lehet bővíteni úgy, hogy csak számokat fogad el. (Pl. 1022 IFA$(CHR$(48) AND A$)CHR$(58) THEN 1020)

Molnár Tibor,
8923 Nemesapáti, Fő u. 17.

```
5 REM: INPUT PRIMORA
10 CLS
20 CLEAR 100:REM:HELYFOGLALAS STRINGEKNEK
30 X=3:Y=2:REM:A MEGJELENITES KOORDINATAI
40 G$="KERDES?":REM:AZ INPUT SZOVEGE
50 GOSUB 1000:REM:INPUT RUTIN HIVASA
60 REM:ITT FOLYTAToDHAT A PROGRAM
70 PRINT"VALASZ:";T$:REM:INPUT TARTALMAT A T$ ADJA
80 END
1000 REM: ******************************
1001 REM: *          INPUT RUTIN          *
1002 REM: *                               *
1003 REM: * BEMENETEK: X,Y - A KERDES     *
1004 REM: * HELYE A KEPERNYoN             *
1005 REM: * G$ - A KERDES SZOVEGE         *
1006 REM: * KIMENET: T$ - A VALASZ        *
1007 REM: ******************************
1010 PRINT$ Y,X,CHR$(6);G$
1015 A$="":T$="":W=0
1020 A$=INKEY$:IFA$="" THEN 1020
1025 BEEP70,100
1030 IFA$=CHR$(13)THEN RETURN
1040 IFA$=CHR$(8)THEN 1100
1050 T$=T$+A$:W=0
1060 PRINT$ Y,X+LEN(G$+T$)+W,CHR$(6);A$
1070 GOTO 1020
1100 IF LEN(T$)<>0 THEN T$=MID$(T$,1,LEN(T$)-1):W=1
:A$=" ":GOTO1060
1105 GOTO 1020
```

botrány

ZX SPECTRUMRA és 2 személyre

Lapocskánk történetében eleddig még nem fordult elő – hál' istennek –, hogy egy közölt program szerzői jogán összevitatkoztak volna olvasóink. Ami késik nem múlik. Íme a furcsa eset dokumentumai:

Előfelvételis katona vagyok Debrecenben, a BIT-LET rendszeres olvasója. Régebben megjelent egy cikkem „Nagyfelbontású grafika ZX 81-en" címmel, azóta „átnyergeltem" Spectrumra. A BIT-LET febr. 26-i számában Rucz Lajos „Rajz a borderen" címmel cikket jelentetett meg. A cikkben közölt egy disassemblált listát a rutinok felhasználásának leírásával, az egyes memóriacímek funkciójával.
Az említett rutint én írtam kb. egy évvel ezelőtt, s BASIC demo-val kiegészítve BORDERPAINT néven ismerhették meg azok, akikhez csere útján eljutott. Rucz Lajos cikkéből hiányoltam, hogy nem említette meg: a rutint egy nem saját programból vette át.
Nem az a célom, hogy a szerzői jogról vitatkozzam, hanem kiegészítsem a megjelent cikket, amely elég hiányos volt (nem említ néhány lehetőséget, amelyet a rutin tud; nincs magyarázat a programlistához; nem említi meg a más memóriacímekre való befordítás problémáit stb.), s így az olvasó önálló kísérletezésének útjába áll. A levelemhez mellékelt cikkben ezeket a hiányosságokat szeretném megszüntetni, s néhány támpontot adni a további fejlesztésekhez.

A háttér színezése ZX Spectrumon

1. Mit csinál a program?

Lehetővé teszi, hogy a borderen egyszerre több színnel „írhassunk": külön színnel a felső keret felső részére (ez a képernyő tetejétől a paper-rész tetejéig tart); valamint a paper-rész két oldalán levő 192 (!) sor mindegyikére. Az egyes háttérrészekre (ezentúl mezők) írhatunk normál vagy villogó színnel. Normál szín: az eddig ismert színek 0–7-ig; villogó szín: az eredeti „flash" ütemének megfelelően (esetleg más fázisban) a normál szín és a kiegészítő szín (kódja: 7–(normál szín kódja)) felváltva jelennek meg.

2. Hogyan csinálja?

A Z80A processzor 2-es megszakítási üzemmódját használja fel: az ULA mindig abban a pillanatban, amikor a képernyőt kezdi el készíteni, megszakítást „kezdeményez", s ezt a Z80 vagy elfogadja, vagy (pl. BEEP, LOAD, SAVE közben) letiltja. Így a képkészítés és a megszakításrutin szinkronban van egymással. A mi saját rutinunk tehát a következőket csinálja: adott idejű késleltetés után az aktuális mező színét (a villogást is figyelembe véve) „írja rá" a borderre, s ezt annyiszor ismétli, ahány mezőnk van.

3. Így épül fel a program:

A lista elején egy ugrótábla található, erre azért van szükség, hogy a fordítási címtől függetlenül könnyen kiszámíthassuk az egyes rutinok hívási címeit. Az INTON rutin bekapcsolja a saját interrupt rutinunkat, az INTOFF kikapcsolja azt. Ebben a megvalósításban az INTRP kezdőcíme csak #FE69=65129 lehet! (INTRP elé ORG 65129 kell.) A feladatot végrehajtó rutin az INTRP. Ez két rutint hív:
1. COLOR: a border alsó és felső részének színét írja a borderre a fázisát figyelembe véve;
2. WAIT: Időhúzás
A következő 5 rutinnak már semmi köze nincs a megszakításhoz, ezek a színmemórián végeznek műveleteket:
1. SCDOWN: lefelé scrolloz
2. SCRUP: felfelé scrolloz
3. INVERT: a színeket invertálja
4. FLON: a villogást bekapcsolja
5. FLOFF: a villogást kikapcsolja
(A rutinokhoz részletes magyarázat található a mellékelt programlistán.)

4. Programbelövés és csapdái

Először a csapdákat említem meg, hogy a program belövésének megértése egyszerűbb legyen:
1. Ha a processzor a #4000–#7FFF címeket írja-olvassa, működése a képernyőkészítéssel összefüggő hardver okokból lassabb lesz. Ebből két dolog következik:
a) A program csak a #8000–#FFFF tartományban helyezhető el (ebben a formájában).
b) Amikor CLEAR utasítással levédjük a memóriát, argumentumként olyan k számot kell megadni, hogy a processzor stack ne kerüljön a „lassú" címtartományba. A k szám értéke a BASIC-ben levő szubrutinhívások mélységétől függően kb. 33000... 65000 címek között kell, hogy legyen.
2. A vektoros megszakítási mód miatt az INTRP rutin kezdőcíme ebben a formában csak 65129 lehet! Amenynyiben a rutint más címre fordítjuk be, „elszáll" a rendszer. Ha mégis szeretnénk más címre befordítani INTRP-t, az INTON rutint át kell alakítanunk:

```
INTON LD HL,INTRP
      LD (#FEFF),HL
      LD A,#FE
      LD I,A
      IM 2
      RET
```

A programbelövés fázisai:
1. Megfelelő assembler/editorral beírjuk a programot (GENS3)
2. Az INTRP elé elhelyezzük az ORG 65129 utasítást (vagy a módosított INTON rutint használjuk
3. A TABLE, INTON, INTOFF programrészeket úgy helyezzük el, hogy sem a színmemóriával, sem az INTRP-vel ne kerüljön fedésbe. (pl.: TABLE elé ORG 65000-t írva)
4. Befordítás, filementés stb.
5. A memória levédése (pl.: CLEAR 64999)
6. Színmemória feltöltése, futtatás

5. A színmemória

Kezdőcímnél a „CMEM EQU..." sorban állíthatjuk be, mérete 192 byte.
A byte-ok bitkiosztása:
b7:0/1: villogás ki/be;
b6–b3: üres;
b2–b0: szín értéke.
Pl. 7: fehér, 135: fekete-fehér vill., 4: zöld, 132: zöld-lila vill.
A legelső byte a legfelső, a legutolsó a legalsó mező színkódja. TOP címre a border felső, BOTTOM címre a border alsó részének színkódja kerül.

6. Érdekességek:

Az alábbi módosításokkal változtathatunk a program működésén, s így érdekes hatásokat érhetünk el:
X1 BIT 4,A/BIT 3,A/BIT 2,A/BIT 1,A/BIT 0,A: a villogás sebességét gyorsíthatjuk

8. Programlista

```
TABLE  JP INTON       ; Az egyes rutinok ugrási
       JP INTOFF      ; táblázata
       JP SCRUP       ;
       JP SCDOWN      ;
       JP INVERT      ;
       JP FLON        ;
       JP FLOFF       ;
INTON  LD A,9         ; Ezzel a rutinnal érjük el, hogy
                        megszakításkor
       LD I,A         ; az INTRP rutin legyen végrehajtva
       IM 2           ; (I*256+255)=>INTRP
       RET            ; Vége
       ;
INTOFF LD A,62        ; Az I regiszter eredeti értékének és az
       IM 1           ; 1-es megszakítási mód
       LD I,A         ; beállítása
       LD A,(23624)   ; Az eredeti border visszaállítása
       AND 56         ;
       RRCA           ;
       RRCA           ;
       RRCA           ;
       OUT (254),A    ;
       RET            ; Vége
       ;
INTRP  DI             ; Amíg a rutin fut, le kell tiltani a
                        megszakítást
       PUSH AF        ; A felhasznált regiszterek elmentése
       PUSH BC        ;
       PUSH HL        ;
       LD A,(COUNT)   ; COUNT-ban az eddigi megszakítások
       INC A          ; minden 16.
       AND 31         ; megszakítás után a villogó mezők
       LD (COUNT),A   ; maszkját változtatjuk, azt a
X1     BIT 4,A        ; MASK-ban tároljuk
X2     LD A,7         ;
       JR Z,INT R1    ;
X3     LD A,Ø         ;
INTR1  LD (MASK),A    ;
       LD A,(TOP)     ; Az A reg. a border felső részének színe
       CALL COLOR     ; => Border
       LD BC,4        ; Várakozás, amíg a border felső
                        részének
       CALL WAIT      ; a végére nem értünk
       LD BC,9473     ;
       CALL WAIT      ;
X4     LD B,192       ; B=mezők száma
       LD A,(MASK)    ;
       LD C,A         ; C=MASK
       LD HL,CMEM     ; HL=a mezők színmemóriájának kezdőcíme
INTR2  PUSH BC        ; Számláló elmentése
       PUSH AF        ;
       LD A,(HL)      ; A színmemória soronkövetkező byte-ja
       BIT 7,A        ; és a MASK segítségével meghatározza a
       JR Z,INTR3     ; kiírandó színt
       XOR C          ;
INTR3  AND 7          ;
       INC HL         ; A színmemória következő byte-jára lép
       OUT (234),A    ; => Border
       LD B,4         ; Várakozás, amíg a következő
INTR4  NOP            ; mező elejére érünk
       NOP            ; A NOP-okat ne hagyd ki!
       NOP            ;
       NOP            ;
       DJN2 INTR4     ;
       POP AF         ;
       POP 3C         ; A számláló visszatöltése
       DJN2 INTR2     : Ciklus. amíg vannak mezők
       LD A,(BOTTUM); A=a border alsó részének színe
       CALL COLOR     ; => Border
       POP HL         ; A felhasznált regiszterek
       POP BC         ; eredeti értékének
       POP AF         ; visszatöltése
       JP 56          ; Ugrás az eredeti int. rutinra
       ;
COLOR  LD B,A         ; B=színkód
       LD A,(MASK)    ; A=MASK
       BIT 7,B        ; Villogás esetén
       JR Z,COL1      ; B=módosított színkód
       XOR B          ;
       LD B,A         ;
COL1   LD A,B         ; A=színkód
       OUT (254),A    ; => Border
       RET            ; Vége
       ;
WAIT   LD A,B         ; Várakozás: B=finom szab.
WA1    LD B,A         ;            C=durva szab.
WA2    DJNZ WA2       ;
       DEC C          ;
       JR NZ, WA1     ;
       RET            ; Vége
       ;
SCDOWN LD HL, CMEM+190 ; NL=színmemória utolsó előtti byte-ja
       LD DE, CMEM+191 ; DE=színmemória utolsó byte-ja
       LD A,(DE)       ; A= az utolsó byte tartalma
X5     LD BC,191       ; BC=mozdítandó byte-ok száma
       LDDR            ; Minden byte-ot eggyel hátrébb mozdít
       LD (DE),A       ; A korábbi utolsó byte lesz az első
       RET             ; Vége
       ;
SCRUP  LD HL,CMEM+1    ; Ugyanaz az elv, mint SCDOWN-nál
       LD DE,CMEM      ;
       LD A,(DE)       ;
X6     LD BC,191       ;
       LDIR            ;
       LD (DE),A       ;
       RET             ; Vége
INVERT LD HL,CMEM      ; HL=színmemória kezdete
       LD B,192        ; A számláló beállítása
INV1   LD A,7          ; Szín fordítása
       XOR (HL)        ;
       LD (HL),A       ;
       INC HL          ; Következő byte-ra lép
       DJNZ INV1       ;
       RET             ; Vége
       ;
FLON   LD HL,CMEM      ; Az elv ugyanaz, mint
       LD B,192        ; az INVERT-nél
FL1    SET 7,(HL)      ; A FLASH-bit 1-be állítása
       INC HL          ;
       DJNZ FL1        ;
       RET             ; Vége
       ;
FLOFF  LD HL,CMEM      ;
       LD B,192        ;
FL2    RES 7,(HL)      ; A FLASH-bit törlése
       INC HL          ;
       DJNZ FL2        ;
       RET             ; Vége
       ;
COUNT  DEFB Ø          ; Számláló
MASK   DEFB Ø          : Maszk
TOP    DEFB Ø          ; Border felső része
BOTTOM DEFB Ø          ; Border alsó része
       ;
CMEM   EQU 65320       ; Színmemória kezdete
```

X2 LD A,N (N=0...7)
X3 LD A,M (M=0...7)
A villogás két állapotának maszkját változtatjuk meg.
X4 LD B,K (K=1...192): a középső rész mezőszámát állítja.
X5 LD BC,NN (NN=1...191): az alsó NN sor scrollozza.
X6 LD BC,MM (MM=1...191): a felső MM sort scrollozza.
Vigyázat! NN és MM más értéke a rendszer „összeomlásához" vezethet!

7. Hogyan tovább?

Akinek sikerült a rutin működésének megértése, önmaga is kísérletezhet egyéb borderhatások elérésével. Az időzítések megváltoztatásával, esetleg elhagyásával nagyobb „háttérfelbontást" is elérhetünk, sőt a border felső részére kicsit durvább ábrákat is rajzolhatunk.

Novák István

Fontosnak éreztük, hogy tisztázzuk a tisztázandókat. Íme Rucz Lajos válasza.

Tisztelt Novák István!

1987. III. 7-én kelt levelére kétségtelenül reagálnom kell. Mélységes elnézést kell kérnem a rutinnal kapcsolatos malőrért, ezért röviden vázolnám az előtörténetet. Kb. 1 évvel ezelőtt egy barátom hibás magnetofonmásolatot juttatott el hozzám, melyen az említett rutin is megtalálható volt – BASIC betöltő és DEMO nélkül. A rutin hibaüzenettel állt le betöltés után, több byte hibásan töltődött be, így rendbe kellett hozni, és ezután lehetett kipróbálni szolgáltatásait. A rutin hamarosan pihenőre vonult, több hasonló társa mellé – melyek közül több is kezeli a BORDER-t, és saját tervezésű. Igen nagy szoftverbázis áll rendelkezésemre, melyek között akad néhány amely esetleg „név" nélkül került hozzám. Így fordulhatott elő, hogy az említett rutin – mely megnyerte a szerkesztőség tetszését – az ön nevének közlése nélkül jelent meg az újságban. A történtekért ezúton még egyszer elnézést kérek.

Egyetértek a kiegészítő információval, azonban annyit mindenképpen meg kell említenem, hogy a rutin mellé összeállított magyarázatok célja elsősorban az volt, hogy a rutin legfontosabb felhasználhatósági paramétereit bemutassák. Tudom, hogy elég nehéz ma megtalálni az igazi nyelvezeti mélységet a felhasználók felé, nem tudni, kinek mi az érthető, ill. mennyi magyarázat szükséges hozzá. Sokan beleesünk abba a hibába, hogy amit tudunk, azt már más is tudja, s ebből bonyodalmak származhatnak. Így úgy érzem, magyarázatai nagyon jól kimunkáltak, értelemszerűek, de több helyen túlságosan az alapok szintjére utaló, márpedig akiről feltételezzük, hogy a gépi kódú programozás szintjéig merészkedik, annak azért illik tudni egy-két dolgot. Az Interrupt-tal kapcsolatos kiegészítéseit szükségesnek tartom. A rutin felépítése szintén csak megerősítési információ, egyébként a gépi kódú rutinból ez a blokkséma jól elkülöníthető. Nagyon jónak tartom, hogy megemlíti a programbelövés csapdáit, s hivatkozik a hardver okokból bekövetkező „lassulás"-ra. Itt azért nem ártott volna hivatkozni arra, hogy ez a hibajelenség megkeserítheti az életünket a LOAD/SAVE műveletek használata esetén is, pl. nem egészséges dolog egy TURBO betöltőt a PRINTER BUFFER területén elhelyezni. A programbelövés útmutatóját jónak tartom, de ez az egy dolog, amelyre, úgy érzem, nincs értelme egységes útmutatót adnunk, hiszen mindenki a jól bevált és megszokott módszerét alkalmazza a cél érdekében. Az érdekességek néhány dologban valóban új információkat nyújtanak az olvasónak, itt én is egyetértek a magyarázat szükségességével. Végül egy rövid megjegyzés. A rutin szándékosan lett decimálisan listázva, elsődlegesen a közvetlen adatbevitel elérése céljából, egyébként a komment mező elsősorban a rutin hosszúsága miatt maradt el.

Rucz Lajos

Bármely program bonyolultsága addig fokozódik, amíg túl nem nő programozója képességein!

(Murphy törvénykönyve)

ÉLET játék

C=64

```
170 REM     ***** ELETJATEK *****
175 REM
180 POKE53280,11:POKE53281,12:PRINT"■■■                    ELET - JATEK■■"CHR$(8)
190 PRINT"          BEALLITAS..■■ @ ■■■■■■+■■■■■■ ? ■■.....JOY2■"
200 PRINT"          RAJZOLAS........... SH":PRINT"          TORLES............. C="
210 PRINT"          RAJZ VEGE.■RETURN■...FIRE■■■■■■■■■■■■■■■■■■■■■■■■■UJ GENERA";
220 PRINT"CIO.......F7■■■■■■■■■■■■■■■■■■■■■■■■FOLYTONOS FEJLODES.F5■■■■■■■■■■■■■";
230 PRINT"■■■■■■■■■■■=''= KIKAPCSOLASA.F3■■■■■■■■■■■■■■■■■■■■■■■■■BEAVATKOZAS.....";
240 PRINT"...F1■■■■■■■■■■■■■■■■■■■■■■■■UJRAKEZDES.........F2":FORI=28672TO28893
250 READA:POKEI,A:NEXTI:PRINT"■■          ■B■ILLENTYU VAGY ■J■OYSTICK"
260 GETA$:IFA$="J"THENJO=1:GOTO280
270 IFA$<>"B"THEN260
280 PRINT"■"
290 K=1024:X=19:Y=12:L=K+40*Y+X
300 O=PEEK(L):POKEL,43:IFJO=1THEN400
310 POKE649,1:POKE650,128
320 P=PEEK(203)
330 IFP=50THENX=X+1
340 IFP=45THENX=X-1
350 IFP=55THENY=Y+1
360 IFP=46THENY=Y-1
```

(330–360: Billentyű figyelés)

```
370 IFP=1THEN590
380 IFP=64THEN320
390 GOTO500
400 P=127-PEEK(56320)
410 IFP=2THENY=Y+1
420 IFP=6THENX=X-1:Y=Y+1
430 IFP=4THENX=X-1
440 IFP=5THENX=X-1:Y=Y-1
450 IFP=1THENY=Y-1
460 IFP=9THENX=X+1:Y=Y-1
470 IFP=8THENX=X+1
```

(410–470: Joystick figyelés)

```
480 IFP=10THENX=X+1:Y=Y+1
490 IFP=16ORPEEK(203)=1THEN590
500 IFX>39THENX=39
510 IFX<0THENX=0
520 IFY>24THENY=24
530 IFY<0THENY=0
540 Z=PEEK(653):M=K+40*Y+X
550 IFZ=0THENPOKEL,O
560 IFZ=1THENPOKEL,81
570 IFZ=2THENPOKEL,32
```

(540–570: SHFT-C= figyelés)

```
580 L=M:GOTO300
590 POKEL,O:R=0:POKE649,10:POKE650,0
600 GETA$
610 IFA$="■"THEN280
620 IFA$="■"THEN290
630 IFA$="■"THENR=1
640 IFA$="■"THENR=0
650 IFA$="■"ORR=1THENSYS28672
660 GOTO600
670 DATA169,0,133,100,169,100,133,101,169,32,162,4,160,0,145,100,200,208
680 DATA251,230,101,202,208,244,169,215,133,98,133,100,169,3,133,99,169,99
690 DATA133,101,169,0,133,251,169,0,133,252,169,0,133,102,160,0,32,176,112
700 DATA162,0,228,252,240,5,160,40,32,200,112,162,39,228,252,240,5,160,42
710 DATA32,200,112,160,80,32,176,112,169,3,197,102,208,9,160,41,169,81,145
720 DATA100,76,124,112,169,2,197,102,208,15,160,41,169,81,209,98,208,7,169
730 DATA81,145,100,76,124,112,160,41,169,32,145,100,32,209,112,230,252,169
740 DATA39,197,252,16,167,230,251,169,24,197,251,16,155,169,0,133,98,133
750 DATA100,169,4,133,99,169,100,133,101,162,4,160,0,177,100,145,98,200,208
760 DATA249,230,99,230,101,202,208,240,96,162,0,228,252,240,3,32,200,112
770 DATA200,32,200,112,162,39,228,252,240,4,200,32,200,112,96,169,81,209
780 DATA98,208,2,230,102,96,230,98,230,100,166,98,208,4,230,99,230,101,96
```

Élet-játék „sorozatunkban" eddig még nem közöltünk C 64-re írt változatot. Vártunk. Tudtuk, hogy előbb-utóbb érkezik egy olyan program, amely gyorsaságban, tudásban fölveszi a versenyt a C 16-os, Spectrumos változattal. Nos, a program megérkezett. Szerzője Solti András Budapestről.

A C 64-re írt változatban a képernyő keretét semlegesnek tekintettem, így az ott levő sejteknek a külső oldalon nincs szomszédjuk. Ezzel a módszerrel elérhető, hogy mind az 1000 cella életképes. A keretet mérgezőnek tekintő változatban az élettér 23∗38 cellára szűkülne.

A program nem nyújt különleges szolgáltatást, a képernyőre berajzolt sejtkonfiguráció életét követhetjük nyomon, lépésenként vagy folyamatosan. Lehetőségünk van a fejlődés során beavatkozásra is, ez főleg a lépésenkénti üzemmódban használható ki.

Ha valakinek igénye van valamely konfiguráció lemezes tárolására, a dr. Uri László: C 64 II. kötetének 9.4 fejezetének végén található programot írja be szubrutinként. Kezdő értéknek 1024-et, végértéknek pedig 2024-et adva. Hívása a 600–660 sorok közt történhet, pl.

605 IF A$="S" THEN GOSUB 2000

Visszatöltés a szokásos „bűvészkedéssel": 150 IF A=1 THEN 290

```
260......:GOTO 271 (280 helyett)
271 PRINT" UJ .....U/Lemezről .....L"
272 GET A$:IF A$="U"THEN280
273 IF A$<>"L"THEN 272
274 INPUT"NEV"; N$ :A=1
275 LOAD N$, 8, 1
```

A BASIC program nem szorul külön magyarázatra. A kezdő állapot beállítása után akár billentyűkkel, akár joystick-kal rajzolhatunk a SHIFT, törölhetünk a C= egyidejű nyomva tartásával.

Az egyszerűbb felépítés, illetve a gyorsabb működés érdekében a billentyűk figyelése a 203 címen található kódok felhasználásával történik. A mozgatásra a d [ ]? gombok szolgálnak.

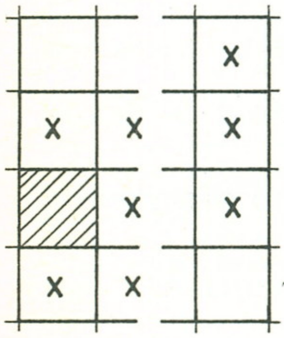
A rajz befejezése után (RETURN vagy tűzgomb) az "F" billentyűkkel választhatunk a többi funkciók között.

Az új nemzedék létrehozása a gépi kódú rutinnal kb. fél másodperc, így folytonos fejlődés esetén látványos sebességgel változik a kép. A további lépésekhez a folytonos fejlődést akkor is ki kell kapcsolni, ha már minden „kipusztult" a képernyőről.

A gépi kódú rutint egy képernyő méretű munkaterület törlése után végigpásztázza a képernyőt. Ha sejt keletkezéséhez, illetve életbenmaradásához alkalmas feltételeket talál, a sejteket a munkaterület megfelelő regiszterébe ülteti. Az utolsó cella vizsgálata után a munkaterületet átmásolja a képernyőre, majd visszatér a BASIC programba.

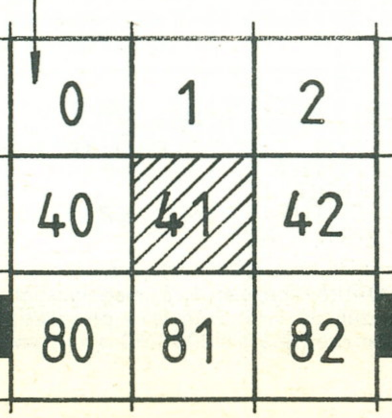
Mivel a képernyő alatti byte-ok értéke alapállapotban 0, a feletti rész (BASIC program eleje) sem tartalmaz 81 kódú karaktert, így nem vizsgáltam, hogy első vagy utolsó sorban van-e a sejt.

Mivel az első, illetve utolsó oszlop szomszédi kapcsolatban van a másik oldal egy sorral eltolt három cellájával, ezek figyelését nem lehet elhanyagolni.



A szomszédok számlálására indirekt indexelt címzést (62), y alkalmaztam. Az index regiszterek csak pozitív eltolást tesznek lehetővé, ezért a pozicionálást a vizsgált cella (41) átlósan bal felső szomszédjára (0) kell elvégezni. (A táblázat az egyes cellákhoz tartozó y értékeket tartalmazza decimálisan.)

Solti András




```
7000  A9 00     LDA #$00
7002  85 64     STA $64
7004  A9 64     LDA #$64
7006  85 65     STA $65
7008  A9 20     LDA #$20
700A  A2 04     LDX #$04
700C  A0 00     LDY #$00
700E  91 64     STA ($64),Y
7010  C8        INY
7011  D0 FB     BNE $700E
7013  E6 65     INC $65
7015  CA        DEX
7016  D0 F4     BNE $700C
7018  A9 00     LDA #$00
701A  85 62     STA $62
701C  85 64     STA $64
701E  A9 04     LDA #$04
7020  85 63     STA $63
7022  A9 64     LDA #$64
7024  85 65     STA $65
7026  A9 00     LDA #$00
7028  85 FB     STA $FB
702A  A9 00     LDA #$00
702C  85 FC     STA $FC
702E  A9 00     LDA #$00
7030  85 66     STA $66
7032  A0 00     LDY #$00
7034  20 AA 70  JSR #70AA
7037  A0 28     LDY #$28
7039  20 B6 70  JSR $70B6
703C  A0 2A     LDY #$2A
703E  20 B6 70  JSR $70B6
7041  A0 50     LDY #$50
7043  20 AA 70  JSR $70AA
7046  A9 03     LDA #$03
7048  C5 66     CMP $66
704A  D0 09     BNE $7055
704C  A0 29     LDY #$29
704E  A9 51     LDA #$51
7050  91 64     STA ($64),Y
7052  4C 70 70  JMP $7070
7055  A9 02     LDA #$02
7057  C5 66     CMP $66
7059  D0 0F     BNE $706A
705B  A0 29     LDY #$29
705D  A9 51     LDA #$51
705F  D1 62     CMP ($62),Y
7061  D0 07     BNE $706A
7063  A9 51     LDA #$51
7065  91 64     STA ($64),Y
7067  4C 70 70  JMP $7070
706A  A0 29     LDY #$29
706C  A9 20     LDA #$20
706E  91 64     STA ($64),Y
7070  20 BF 70  JSR $70BF
7073  E6 FC     INC $FC
7075  A9 25     LDA #$25
7077  C5 FC     CMP $FC
7079  10 B3     BPL $702E
707B  20 BF 70  JSR $70BF
707E  20 BF 70  JSR $70BF
7081  E6 FB     INC $FB
7083  A9 16     LDA #$16
7085  C5 FB     CMP $FB
7087  10 A1     BPL $702A
7089  A9 00     LDA #$00
708B  85 62     STA $62
708D  85 64     STA $64
708F  A9 04     LDA #$04
7091  85 63     STA $63
7093  A9 64     LDA #$64
7095  85 65     STA $65
7097  A2 04     LDX #$04
7099  A0 00     LDY #$00
709B  B1 64     LDA ($64),Y
709D  91 62     STA ($62),Y
709F  C8        INY
70A0  D0 F9     BNE $709B
70A2  E6 63     INC $63
70A4  E6 65     INC $65
70A6  CA        DEX
70A7  D0 F0     BNE $7099
70A9  60        RTS
70AA  20 B6 70  JSR $70B6
70AD  C8        INY
70AE  20 B6 70  JSR $70B6
70B1  C8        INY
70B2  20 B6 70  JSR $70B6
70B5  60        RTS
70B6  A9 51     LDA #$51
70B8  D1 62     CMP ($62),Y
70BA  D0 02     BNE $70BE
70BC  E6 66     INC $66
70BE  60        RTS
70BF  E6 62     INC $62
70C1  E6 64     INC $64
70C3  A6 62     LDX $62
70C5  D0 04     BNE $70CB
70C7  E6 63     INC $63
70C9  E6 65     INC $65
70CB  60        RTS
```

SORVEZETŐ

A B C D E F G H I

A Bolyai János Matematikai Társulat Informatikai Bizottsága (BJMTIB) és az Országos Pedagógiai Intézet Matematika Osztálya közösen készített egy füzetet, amelyben többek között olyan HT-programok vannak, melyek segíthetnek a matematika oktatásában. Ezek közül többet „átfogalmazunk'' C16, illetve C+4 gépekre, és néhány folytatásban közöljük az alábbi csoportosításban:

- **Prímek, törzstényezők, legnagyobb közös osztó**
- **Függvényábrázolás**
- **Kombinatorika, számolás nagy számokkal**
- **Rendezési algoritmusok**
- **Keverés, véletlen**
- **Grafika, SHAPE-ek**

Előzményként említhetjük a BJMT–KFKI közös kiadásában a BASIC Példatár II. kötetét; a Kőhegyi János és munkatársai szerkesztette „Programozási forgácsok''-at a Számítástechnika folyóiratban 1984 körül; Kovács Mihály és Tarcsay Géza írásait a Technika újságban 1983–84. években; és legújabban Simonovits Miklós: Számítástechnika c. tankönyvét. A folyóiratok könyvtárakban elérhetők, ezért a programok megismétlése helyett csak hivatkozni fogunk rájuk.

A szinalányomással jelölt feladatok megoldásai beküldhetők. Ebből egyrészt az anyagok visszhangjára szeretnénk következtetni, másrészt sok kérdésre magunk sem ismerjük az egzakt feleletet. A helyes (és frappáns) válaszok beküldői között apróságokat (floppy, kazetta, BITLET-ek stb.) sorsolunk ki.

PRÍMEK, OSZTHATÓSÁG, LNKO

1. feladat: Írjuk ki 1 és N között a prímszámokat!

a. megoldás:

```
4 SCNCLR
5 INPUT "EGYT0L MEDDIG IRJAM KI A PRIMEKET";N
10 M=INT(SQR(N)):M1=N/5:IF M1<100 THEN M1=100
20 DIM A(M1)
80 K=1:A(K)=2:PRINT 2;
100 FOR I=3 TO N
110 FOR J=1 TO K
120 IF INT(I/A(J))=I/A(J) THEN J=K+3
130 NEXT J
140 IF J<K+2 THEN K=K+1:A(K)=I:PRINT I;
150 NEXT I
170 PRINT:PRINT "EZ";100*K/N;"SZA´ZALE´K!"
```

Tipikus rekurzió: a K-adik prímhez (A(K)) az előzőkön keresztül vezet az út. Ezért kell az elsőt (A(1)=2 a 80. sorban) külön kezelni. Figyeljük meg a 120. és 140. sorokat: „Nincs olyan prím, amivel osztható'', és „Van olyan…'' esetek szétválasztása így korrekt és gyors.

b. megoldás:

Nagyobb N értékekre időt takarítunk meg, ha a 110. sorban a ciklusváltozó végértékét K és M=SQR(N) közül a kisebbiknek választjuk:

```
110 FOR J=1 TO (K+M-ABS(K-M))/2
```

avagy

```
105 K1=K:IF K1>M THEN K1=M
110 FOR J=1 TO K1
```

c. megoldás:

Eratoszthenesz szitáját használjuk:

```
4 SCNCLR
5 INPUT "EGYT0L MEDDIG IRJAM KI A PRIMEKET";N
20 DIM A(N)
80 L=1:K=2:PRINT 2;
100 FOR I=K TO N STEP K:A(I)=1:NEXT I
120 K=K+1:IF K>N THEN 170
150 IF A(K)=1 THEN 120 : ELSE PRINT K;:L=L+1:GO TO 100
170 PRINT:PRINT "EZ";100*L/N;"SZA´ZALE´K!"
```

A rekurzió újabb szép példája. Egyetlen FOR ciklus van, a „másik'' rejtett: a 70. és a 120. sorokban egy IF-es szervezés fedezhető fel. Az előzőkhöz hasonlóan itt is gyorsíthatunk, bár ez bonyolultabbnak is tűnhet:

```
4 SCNCLR
5 INPUT "EGYT0L MEDDIG IRJAM KI A PRIMEKET";N
20 DIM A(N)
100 FOR I=2 TO SQR(N)
110 IF A(I)=0 THEN PRINT I;:L=L+1:FOR K=I*I TO N STEP I:A(K)=1:NEXT K
120 NEXT I
150 FOR I=INT(SQR(N)+1) TO N:IF A(I)=0 THEN PRINT I;:L=L+1
160 NEXT I
170 PRINT:PRINT "EZ";100*L/N;"SZA´ZALE´K!"
```

Okvetlenül hasonlítsuk össze az eddigi négy programnak a futási idejét! N=500 esetén már jelentős különbségek tapasztalhatók.

2. feladat: Az ikerprímek (különbségük pontosan kettő) inverzben jelenjenek meg.

```
4 SCNCLR
5 INPUT "EGYT0L MEDDIG IRJAM KI A PRIMEKET";N
20 DIM A(N+2)
30 L=1:A(1)=1:A(N+1)=1:A(N+2)=1:PRINT 2;
100 FOR I=2 TO SQR(N)
110 FOR K=I*I TO N STEP I:A(K)=1:NEXT K
120 NEXT I
150 FOR I=3 TO N STEP 2:IF A(I)=0 THEN GOSUB 200
160 NEXT I
170 PRINT:PRINT "EZ";L;"DARAB, AZAZ";100*L/N;"SZA´ZALE´K!"
180 PRINT "EBB0L";1+INT(LL/2+.01);"IKERPRIM!"
199 END
200 IF A(I-2)*A(I+2)=0 THEN PRINT "▩";:LL=LL+1
210 L=L+1:PRINT I;"■";
290 RETURN
```

3. feladat: Írassuk ki A és B között a prímeket!

Nyilván nincs szükség általában az A előtti összes prímre, csak B-ig kell megkeresni a törzsszámokat.

```
4 SCNCLR
5 PRINT "METTOL MEDDIG IRJAM KI A PRIMEK
ET (KE´T"
6 PRINT "SZA´MOT IRJ VESSZOVEL ELVA´LASZ
TVA)!":INPUT A,B
10 N=B-A+1:S=SQR(B):MM=S/5:IF MM<100 THE
N MM=100
20 DIM A(N),B(MM)
100 FOR I=2 TO SQR(S)
110 IF A(I)=0 THEN L=L+1:B(L)=I:FOR K=I+
I TO S STEP I:A(K)=1:NEXT K
120 NEXT I
150 FOR I= INT(SQR(S)+1) TO S:IF A(I)=0
THEN L=L+1:B(L)=I
180 NEXT I
200 FOR I=A TO B:A(I-A)=I:NEXT
250 FOR J=1 TO L:FOR K=A TO A+B(J):IF K/
B(J)=INT(K/B(J)) THEN 265
260 NEXT K
265 IF K=B(J) THEN K=K+K
270 FOR I=K TO B STEP B(J):A(I-A)=0:NEXT
I
280 NEXT J
290 PRINT:PRINT TAB(8);A;" ES ";B;" KOZO
TTI PRIMEK:":PRINT
400 FOR I=0 TO B-A:IF A(I)=0 THEN 415
410 PRINT A(I);:P=P+1
415 NEXT I
420 PRINT:PRINT" EZ ";P;" SZAM,AZAZ "(10
0*P)/N;" %"
```

A 100–180. sorok az 1. c. megoldásból vehetők, csak most meg is jegyezzük az 1-től B-ig levő prímeket (B(L)). A feladat tulajdonképpeni megoldása a 200–280. sorokban van. A 250. sorban keressük

meg az [A, B] számintervallum legkisebb elemét, amelyik osztható a J-edik prímmel, és ettől kezdve „szitálunk". A 265. sor tekintettel van az A⟨B esetre.
További jelentős gyorsítás nagy A értékek esetén, ha két sort módosítunk:

```
250 FOR J=1 TO L
260 K=B(J)*INT(A/B(J)):IF K<A THEN K=K+B(J)
```

Megtehetjük, hogy kiindulásként kiírjuk a képernyőre az [A, B] számköz számait, és ezek közül eltüntetjük az összetetteket. Erre egy egyszerűbb változatot mutatunk, ahol feltételezzük, hogy B⟨10000! Előző programunkhoz kell az alábbi módosításokat hozzáadni!

```
195 SCNCLR
200 FOR I=A TO B:A(I-A)=I:GOSUB 500:NEXT
240 W=1
250 FOR J=1 TO L
260 K=B(J)*INT(A/B(J)):IF K<A THEN K=K+B
(J)
265 IF K=B(J) THEN K=K+K
270 FOR I=K TO B STEP B(J):A(I-A)=0:GOSU
B 500:NEXTI
275 GETKEY A$
280 NEXT J
399 END
500 I1=I-A:J1=INT(I1/ 9):I2=2+4*(I1-J1*
9)
510 IF W=1 THEN CHAR 1,I2,J1,"      ":GO T
O 590
520 CHAR 1,I2,J1,MID$(STR$(A(I1)),2,4)
590 RETURN
```

Gombnyomásra (275. sor) tűnnek el az egyes prímek többszörösei.
4. feladat: Bontsunk törzstényezőire egy tetszőleges számot! Először vezessük vissza egy már megoldott feladatra a jelenlegit! Az utolsó listából meghagyva a 4. és 100–180. sorokat, rendelkezünk is a szóba jöhető tényezőkkel, csak ki kell válogatni a valóban megfelelőket. Bővítve néhány további sorral, készen is vagyunk:

```
4 SCNCLR
5 INPUT "MELY SZA'MOT BONTSUK FEL TE'NY.
";B
10 N=B:S=SQR(B):MM=S/5:IF MM<100 THEN MM
=100
20 DIM A(S),B(S)
100 FOR I=2 TO SQR(S)
110 IF A(I)=0 THEN L=L+1:B(L)=I:FOR K=I+
I TO S STEP I:A(K)=1:NEXT K
120 NEXT I
150 FOR I= INT(SQR(S)+1) TO S:IF A(I)=0
THEN L=L+1:B(L)=I
180 NEXT I
190 K=L:L=0
195 SCNCLR
200 FOR I=1 TO K-1
210 IF INT(B/B(I)+.01)=B/B(I) THEN L=L+1
:A(L)=B(I):B=INT(B/B(I)+.01):GO TO 210
220 NEXT I
230 IF B>1 THEN L=L+1:A(L)=B
250 PRINT:PRINT
300 IF L=1 THEN PRINT B;" PRIMSZA'M.":EN
D
400 PRINT N;"=";A(1);:FOR I=2 TO L:PRINT
 "*";A(I);:NEXT I
```

A 210. sorban találhatunk osztó tényezőt, osztunk is vele, és csak a hányadost vizsgáljuk tovább. Kiíratáskor (300. és 400. sorok) meg kell különböztetni, prím-e a szám, vagy vannak valódi osztói.
Megjegyzés: A Technika 1984. II. számában szerepel egy egyszerűbb változat, ami prímkeresés nélkül dolgozik. Kivonatosan közöljük ezt a listát is:

```
10 SCNCLR
50 INPUT "MELY SZA'MOT BONTSUK FEL";A
60 B=LOG(A)/LOG(2)+2:DIM V(B)
110 X=A
120 B=INT(A/2)+1
130 B=B-1:C=A/B
140 IF INT(C)<>C THEN 130
150 Q=Q+1:V(Q)=C
160 IF B<>1 THEN A=B:GO TO 120
170 PRINT "A TE'NYEZ0K: ";:FOR I=1 TO Q:
PRINT V(I);: NEXTI
```

Módosítsuk egy kicsit:

```
10 SCNCLR
50 INPUT "MELY SZA'MOT BONTSUK FEL";A
60 B=LOG(A)/LOG(2)+2:DIM V(B)
110 X=A
120 B=1
130 B=B+1:C=A/B
140 IF INT(C)<>C THEN 130
150 Q=Q+1:V(Q)=B
160 IF C<>1 THEN A=C:B=B-1:GO TO 120
170 PRINT "A TE'NYEZ0K: ";:FOR I=1 TO Q:
PRINT V(I);: NEXTI
```

5. feladat: Euklideszi algoritmus segítségével adjuk meg két szám LNKO-ját! Használjuk fel, hogy A és B közös osztója az ABS(A–B)-nek is osztója.

```
10 INPUT A,B
90 IF A>B THEN C=A:A=B:B=C
100 IF B=A THEN PRINT A:END
200 B=B-A:IF B>A THEN 200 :ELSE 90
```

Egy tömörebb, de talán kevésbé triviális változat:

```
10 INPUT A,B
20 C=(A+B-ABS(A-B))/2:A=ABS(A-B):B=C
30 IF A=B THEN PRINT B :ELSE 20
```

Végül jelentős gyorsítást eredményez:

```
10 INPUT A,B
20 IF A>B THEN C=A:A=B:B=C
30 B=B-INT(B/A)*A:IF B=0 THEN PRINT A :E
LSE 20
```

További feladatok:

1. Általánosítsuk a 8. lista programját: tetszőleges (ésszerű) korlátok között is „optimálisan férjenek el" a számok a képernyőn!

2. Vajon mi okozza (mik okozzák) az első négy program sebességei közötti különbségeket?

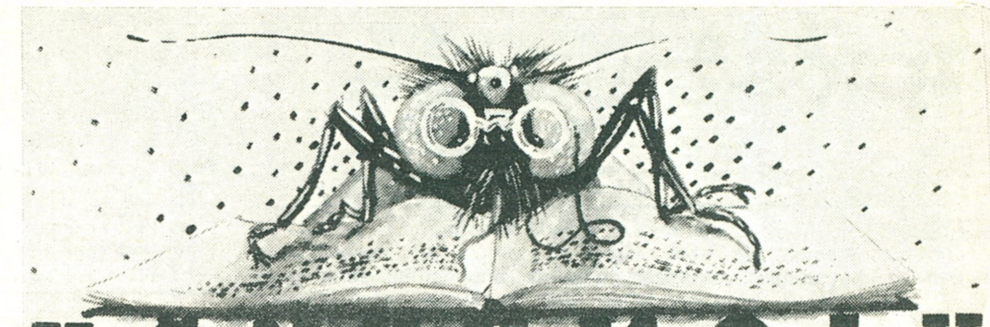
3. A tényezőkre bontást is három különböző programmal próbáltuk. Hasonlítsuk össze ezeket is futási idő szempontjából! Itt mi okozza az eltéréseket?

4. Írassuk ki a törzstényezőkre bontást hatványalakban!

5. Tényezőre bontással határozzuk meg két szám LNKO-ját, illetve LKKT-jét! Mi a véleményed erről, összevetve a megfelelő listákkal?



Rácz Mihály–Horváth Attila: FiziKomp – Novotrade, 227 o., 129 Ft
(A szerzők a közegellenállás, a rezgőmozgás és a gravitáció témaköréből oldanak meg feladatokat BASIC és Pascal nyelven. Részletesen ismertetik a Pascal programnyelv használatát a C–64-en és Spectrumon is.)
Valkó Péter–Vajda Sándor: Műszaki-tudományos feladatok megoldása személyi számítógéppel – Műszaki Könyvkiadó, 342 o., 90 Ft
(A kötet a mérnöki munkában gyakran előforduló feladatok megoldását mutatja be IBM, illetve azzal kompatibilis gépeken.)

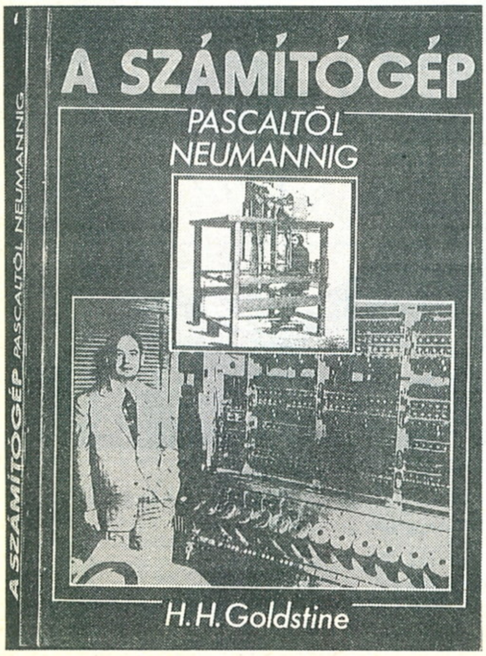


KÖNYVMOLY

H. H. Goldsteine: A számítógép Pascaltól Neumannig – Műszaki Könyvkiadó, 368 o., 110 Ft

Herman H. Goldstine a számítástechnika, a számítógéptudomány élő klasszikusa. Vezető szerepet játszott az ENIAC (Electronic Numerical Integrator and Computer), az első elektronikus számítógép kifejlesztésében, amely elődeitől eltérően nem tartalmazott mechanikus működésű alkotóelemeket. Így ez a gép tekinthető az első, mai értelemben vett számítógépnek. Oroszlánrészt vállalt az EDVAC létrehozásában is, amely a modern számítógépek szinte valamennyi ismérvével rendelkezett. Ami pedig számunkra különösen fontos lehet: közvetlen munkatársa és jó barátja volt Neumann Jánosnak, aki a számítógépek működésének elméletét alkotta meg.

A Műszaki Könyvkiadó Neuman halálának 30. évfordulója alkalmából adta ki Goldstine kötetét, a Neumann János Számítógép-tudományi Társasággal együttműködve. Talán nem kellett volna ilyen sokáig várni a könyv magyar megjelentetésével – ami jó 15 éve került először az amerikai olvasók kezébe –, de Goldstine számítógép-történeti beszámolója ma is nagy érdeklődésre tart számot mind a számítógépek, mind a technikatörténet iránt érdeklődők körében. A kiadásért mindenesetre elismerés illeti Kovács Győzőt, a kötet lektorát, a Neumann János Számítógép-tudományi Társaság elnökét.



A könyv rengeteg hasznos információt közöl olyan kitűnő stílusban, hogy szinte lehetetlen letenni. Mint a szerző az utolsó fejezetben említi, a számítógép-tudomány fejlődését Pascal születésétől, 1623-tól Neumann haláláig – 1957-ig – tekintette át. Ha mindehhez még azt is hozzászámítjuk, hogy műve elején a Pascal munkásságát megelőző eredményeket is számba veszi, akkor nyilván felvetődik a kérdés: hogy fér el mindez alig több mint 300 oldalon?

Nos, Goldstine-nak úgy sikerült mondanivalóját a terjedelmi határok közé sűrítenie, hogy az olvasó közben sohasem érzi azt, hogy egy agyontömörített, csak a bennfentesek számára érthető szakmunkával kell megbirkóznia.

A könyvben teljesen természetes módon váltogatják egymást az egyes gépek működési módját bemutató szakmai szintű leírások, a matematikai kifejezések, képletek és a gépek megalkotóit jellemző rövid anekdoták. Mindez egyáltalán nem zavaró, mert egységes egésszé ötvözi a szerző stílusa. Összefüggő történetet, szinte regényt olvashatunk attól kezdve, hogy Babbage ötlete alapján – aki a mechanikus számolási műveleteket kívánta gépesíteni az akkori csúcstechnológiát képviselő gőzgépek segítségével – megszületett az első differenciagép, egészen a modern digitális, bináris számokat használó gépek megalkotásáig.

A munka három fő részre tagolódik. Az első a II. világháború előtti tudományos eredményekkel foglalkozik, a második a háború idején végzett fejlesztési munkákkal, a harmadik pedig a háború befejeződése és az 1957 közötti időszak eseményeivel. Mindhárom korszak nagyjából azonos terjedelmet kap a könyvben – ebből adódik, hogy az 1942 és az 1946 közötti évekkel foglalkozik a legrészletesebben. Ennek oka nyilvánvaló: a hadiiparnak hatalmas szüksége volt a fegyverek ballisztikai számításait gyorsan elvégző gépekre – ez volt a számítógépek kifejlesztésének „aranykora".

Goldstine hatalmas információs anyagot dolgozott fel. Erre utal, hogy a kötet minden oldalán három-négy lábjegyzet olvasható, amelyek vagy könyvekre, cikkekre hivatkoznak, vagy – hiszen a több évtizedes emlékek nem feltétlenül megbízhatóak – saját, kiterjedt levelezésére. Beszámolója is akkor válik igazán átfűtötté, amikor már nem az előzményekről, a múltról beszél, hanem arról az időszakról, amelyben már ő maga és barátja, Neumann is részese a számítógép-tudomány fejlődésének. E fejezetekben akár önhittnek is hihetnénk a szerzőt, hiszen szinte minden második oldalon leírja a saját nevét – de nem tehet mást, elvégre tevékeny szerepet játszott az eseményekben, és amikor felsorolja, hogy kik tevékenykedtek egy-egy feladat megoldásán, nem tagadhatja le saját részvételét.

Kifogásaink csak a kötet szerkezetét illetően lehetnek – és ezek is inkább kötekedésnek tekinthetők.

– Goldstine lendületes beszámolójában sokszor felborul az időrend. Egy-egy gép vagy részegység fejlesztését ismertetve időnként túlságosan előreszalad, majd nem győz visszatérni a kiindulási időponthoz. Így, ha nem figyelünk eléggé a dátumokra, nem tudjuk meg, hogy mi volt előbb és mi később, vagyis hogy mi minek az eredménye.

– Máskor viszont nem követi azt a szerkesztési elvet sem, hogy egy-egy részterületet az elejétől a végéig bemutatna. Így számtalan bekezdés végén olvashatjuk, hogy „Erről a későbbiekben részletesen beszámolunk." Így az olvasó nem győz oda-vissza lapozni.

– Goldstine a 118. oldalon Atanasoff naivitásával kapcsolatban említi, hogy egyik munkájában hosszasan értekezett a számok egyik számrendszerből a másikba való átváltásáról, és nem ismerte fel, hogy ez mennyire triviális. Nos, Goldstine is beleesik ebbe a hibába: matematikai fejtegetései korántsem azonos színvonalúak. Hol tel-

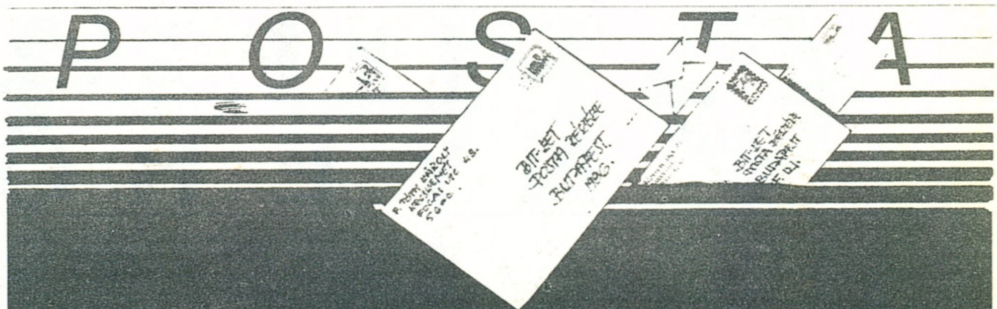
jesen elemi dolgokat elemez hosszasan, hol pedig bonyolult tételeket tekint magától értetődőeknek.
Még néhány szó a kötet három függelékéről.
Az elsőt maga Goldstine illesztette művéhez, bemutatva az európai országok számítástechnikájának fejlődését.
A második Kovács Győző, a lektor munkája, ez a magyar számítógéptudomány és -gyártás eredményeit ismerteti a hatvanas évek közepéig. Ami a legszembeötlőbb, az az, hogy már a számítógép-tudomány kezdeti korszakában is tízéves hátrányban voltunk a fejlett országokhoz képest – és ez a hátrány fokozatosan nőtt.
A legfurcsább a harmadik függelék. Ez Nicolas Vonneumann-nak, Neumann János fiának kiigazítását tartalmazza. A hibák, amiket Goldstine beszámolójában javít, nem érdemiek. Úgy tűnik, hogy Nicolas apja erkölcseit és világnézetét igyekszik rózsaszínűre festeni. Neumann századunk egyik legnagyobb tudósa volt – nem hisszük azonban, hogy szüksége lenne a szentté avatásra.

Tallér József

Kritikus szemmel

Egy vagyok a magyarországi spectrumosok táborából. Közülük is egy a megszállottak közül, aki megpróbál a gép lelkéhez közel férkőzni; ez alatt a gépi kódú programozás rejtelmeinek megismerését értem. Éppen ezért vettem örömmel észre **Meggyesházi János–Pintér Tibor: ZX SPECTRUM haladóknak, Felhasználási segédlet** c. könyvét.
A könyv elejével meg vagyok elégedve: a rendszerváltozók leírása, az assemblerek használati utasítása hasznos dolog. Utána jönne az érdemi rész: gépi kódú rutingyűjtemény.
Első gépi kódú program: PRG(2). átsorszámozás. A 007A sorában pedig a hivatkozás: JP NZ,9312. Keresem a 9312-es címkét, sort, címet, sehol sincs. A programról annyit írnak, hogy ott van a 118. oldalon. Lapozzunk tovább... PRG(6). Programtömörítés 00B4 címen JP Z,91FB. A helyzet ugyanaz, ne is keress hivatkozást, hogy hol van, mi az a 91FB cím. Ugorjunk a PRG(9). Tárinformációk kiíratása c. részre. Egy kis figyelemmel felderíthetünk egy durva hibát: a veremműveletek nincsenek egyensúlyban: 2 db PUSH után 3 db POP jön. Ettől független begépeltem a programot. Nem szállt el, mint ahogy vártam, de nem is működött. Folytathatnám az elvi hibás programok felsorolását, az ismeretlen vagy nem létező címkére ugró utasítást végrehajtó programokat.
Ne is vizsgáljuk tovább a programok megvalósítását! Nézzük, milyen programokat ajánlanak, írnak le a szerzők! Különböző képernyő scrollok: LORES, HIRES, jobbra, balra, le, fel. Erről csak annyit, hogy jobban jár az ember fia, ha ezeket a rutinokat Rucz Lajos: Rutinról rutinra (avagy bepillantás a Sinclair Spectrum gépi kódú programozásába) c. könyvéből nézi meg. (Esküszünk, hogy nem szándékos, hogy plágiumon ért szerzőnkről ugyanabba a lapszámba dicsérő sorok is bekerültek.) Bár az említett könyv csaknem kétszer annyiba kerül, de a programokhoz magyarázat van, és a programok futásképesek. Úgy látszik, fele pénzért csak az ötletet kaphatjuk meg, hogy tökéletesen megvalósítva legyen leírva, ahhoz, úgy látszik, fizetni kell. És végső soron fizetni kell annak is, aki megvette a ZX Spectrum haladóknak c. könyvet, és csaknem semmire nem tudja használni. Úgy is mondhatnám, hogy nem azt kapja az ember a pénzéért, amit várt.

Mező Gyula
1117 Bp., Irinyi J. u. 42/1806



Tisztelt Szerkesztő!
A BIT-LET-ről jut eszembe: kellene csinálni egy „mérleget". Pl.: így nézne ki:

A BIT-LET mérlege

Jó tulajdonság	*Rossz tulajdonság*
Olcsó	*Mellékletként tengődik*
Tartalmas	*Kevés a spectrumanyag*
Nem ír dirr-durr játékról	*Ritkán jelenik meg*
Van ingyenes hirdetési rovat	*Megjelenés utáni napon már nem lehet kapni*
Átfogó: a hírektől a szoftver mütyürökig mindenről szól	*Nincs benne játékprogram **
Gépnyerő	*Nincs játékprogramteszt (egyszer volt: max. 90 perc: Menekülés, Ki jut az erdőbe?)*
stb.	*stb.*

** Ezt az olvasók írták, keményen el lettek utasítva. Mondjuk egyetértek vele, de az alapelv...*

Most egy kér(d)és: mondjuk írtam egy játékot, és színvonala elérte az ismerősök nemcsak udvariasságból mondott elismerését. Nem kell a MASTERTRONIC, elég lesz a NOVOTRADE is mecénásnak. Hol találom ő(ke)t? Ha ezt, vagy más ilyesmi „szoftver" címét tudja, kérem írja meg! Telefonszám nem jó, mert nincs telefonom (akkor ők hívnak majd fel engem – ez csak vicc volt). És hogy történik egy ilyen vásárlás? Kölcsönös szimpátia esetén? (Novotrade-nak tetszik a program, nekem a Novotrade ajánlata?) A játékról csak annyit, hogy nincs még kész, ír a borderre és zenél (KRAFTWERK). Még egy kérdés: ha odaadom (elfogadják) a Novotrade-nak tegnapelőtt, holnapután a boltban látom?

Zalka Ernő
(MCB) 9022 Győr, Árpád út 51/A II. 14.

Ui.: Tényleg, megvan még a C 16-ja?
Válaszaink sorban visszafelé:
1. A C 16-ot lecseréltem egy Plusz/4-re.
2. Az „alku" valahogy úgy megy, ahogy ön leírja. Azt azonban ne remélje, hogy egy megvásárolt szoftver máról holnapra az üzletbe kerül.
3. A cégeket, amelyek szoftvert vásárolnak egyszerűen föl lehet keresni (utcáról be kell menni) és fölajánlani az árut.
Íme kapásból 1-2 cím:
Novotrade–Deltasoft – Bp. XIII., Kresz Géza u.
Novotrade–Caesarsoft – Bp. III., Szentendrei út
Newline vállalkozás – Vecsés, Diófa u. 15.
ÁPISZ (Áruforgalmi Osztály) – Bp. VII., Wesselényi u.
4. Ami a kis táblázatot illeti jól szórakoztunk rajta. A játékprogramokkal kapcsolatos megjegyzésén elgondolkodtunk. Az az igazság, hogy kellemes, nem túl nagy méretű, s lehetőleg nem piff-puff játékot szívesen közölnénk, ha kapnánk. (Lásd Életjáték) A játékprogram fent nem nálunk jelent meg hanem a Commodore újságban! Üdvözlettel: A szerkesztő

PLUS/4 NYERŐ

A 3. FELADAT MEGOLDÁSA

1. feladat: Lehetséges, bármilyen páratlan számú induló esetén elérhető, hogy mindenkinek ugyanannyi győzelme és vesztesége van. Az állítást a teljes indukció módszerével bizonyíthatjuk legegyszerűbben: 3 versenyzőre az állítás nyilván igaz. Legyen n páratlan szám, s tegyük fel, hogy n versenyző esetére már tudjuk, hogy igaz az állítás, azaz megadhatók olyan páronkénti eredmények, hogy mindenkinek ugyanannyi győzelme és veresége van. Ekkor nyilván mindenkinek $\frac{n-1}{2}$ győzelme, és ugyanannyi veresége van. Be szeretnénk látni, hogy n+2 versenyző esetén is elérhető, hogy mindenkinek ugyanannyi győzelme és veresége legyen. Válasszunk ki 2 versenyzőt (Első Elemért és Második Mártont), s a páronkénti eredményeket a következőképpen határozzuk meg: a többi versenyző között a feltételezésünk miatt meg lehet olyan eredményeket határozni, hogy egymás között mindenkinek $\frac{n-1}{2}$ győzelme és veresége legyen. Válasszunk ki közülük $\frac{n+1}{2}$ versenyzőt, s mondjuk azt, hogy ezek megverték Elsőt, de kikaptak Másodiktól, a többiek viszont Másodikat verték meg, s Elsőtől kaptak ki. Ezután, ha rögzítjük, hogy Első Elemér megverte Második Mártont, könnyű kiszámolni, hogy valóban mindenkinek pontosan $\frac{n+1}{2}$ veresége, s ugyanannyi győzelme van. Ezzel állításunkat beláttuk. (3-ra igaz, s a módszer mutatja, hogy akkor 5-re is, de akkor 7-re is, s így tovább, 99-re is, s í. t.)

2. feladat: A kiírásba – immár hagyományosan – kisebb hiba csúszott, a szöveg második felében a 99 helyett 100 szerepel. Mivel a 2. állítás akárhány versenyzőre igaz, ez nem zavarhatta túlságosan a megoldókat, természetesen elfogadjuk azt is, ha valaki 99-re, s azt is, ha 100-ra bizonyította. (Tervezzük, hogy ezentúl esetleg minden feladat 2 feladatból fog állni:

1) Keressék meg a feladat szövegében a hibét, s javítsák ki, hogy a feladat értelmes legyen, s az állítása igaz.

2) Oldják meg a kijavított feladatot!

Visszatérve a mostani feladatra:

Állítás: akárhány (n) versenyző és akármilyen eredmények esetén lehet a versenyzőket úgy rangsorolni, hogy mindenki legyőzte a rangsorban közvetlenül utána állót.

Bizonyítás: ha néhány versenyzőt sorba tudunk rendezni a feladatnak megfelelően, ezt a részsorbarendezést hívjuk láncnak. Egy lánc hossza a benne szereplő versenyzők száma. (Ha van legalább 2 induló, akkor biztos van 2 hosszú lánc, bármely 2 versenyzőt tud láncot alkotni a megfelelő sorrendben.) Válasszunk ki egy lehető leghosszabb láncot! Ha a lánc hossza n, akkor készen vagyunk. Ha rövidebb, akkor van egy versenyző, Lánconkívüli Lajos, aki nem szerepel a láncban. Lajos nyilván kikapott a lánc első versenyzőjétől, de megverte az utolsót, hisz különben a lánc valamelyik végéhez hozzá lehetne „ragasztani", s így egy hosszabb láncot kapnánk, ami viszont ellentmond a lánc választásának. Mivel Lajos kikapott a lánc első versenyzőjétől, de megverte az utolsót, kell lenni a láncban két egymást követő versenyzőnek (Iedik Imre és Ipluszegyedik Imre), akik közül Lajos Iediktől kikapott, de Ipluszegyediket megverte. Ekkor azonban képezhetjük a következő láncot: vegyük sorra az eredeti láncból az első, második stb. versenyzőt egészen Iedik Imréig, utána vegyük be Lajost, majd Ipluszegyedik Imrétől folytassuk az eredeti láncot végig. Így megint egy eggyel hosszabb láncot kaptunk, ami ellentmond a lánc választásának. Így a leghosszabb láncunk csakis n elemű lehet, s ez a lánc adja a kívánt sorrendet.

Ezzel (legalábbis egy időre) a gráfelméleti jellegű feladatok végére értünk. A gráfok iránt érdeklődőknek ajánljuk Andrásfai Béla: Ismerkedés a gráfelmélettel c. könyvét.

COMPUTER-S ATARI NYERŐ

2. FELADAT

Egy BASIC program a következő sorral kezdődik: 10 INPUT H,N Azt szeretnénk, hogy a program az 1987-es év bármely napjáról kiírja, hogy az a hét melyik napjára esik.
H jelenti a hónap sorszámát (1–12), N a napét (1–31). A beírt adatokat nem kell ellenőrizni.
BASIC-ünkről annyit kell tudni, hogy 1 programsor akármilyen hosszú lehet, s hogy az összehasonlítások értéke –1, ha igaz és 0, ha nem igaz. A BASIC-ünk ismeri a MID $(string, kezdet, db) függvényt.
Feladat, hogy a kívánalmaknak megfelelően fejezzük be a programot. Tehát, ha pl. az input 2, 18, akkor az output SZERDA legyen. Azonban a cél az, hogy ezt minél kevesebb BASIC utasítással (nem programsorral!) érjük el! Az értékadás is utasításnak számít! Előre eláruljuk, hogy a feladat egyetlen PRINT utasítással megoldható!

COMPUTER-S
Kérjük levágni és a levélre felragasztani! Beküldési határidő: 1987. május 26.

BIT-LET Szerkesztőségünk címe: ÖTLET Budapest 1986. Szerkesztő: Angyalosi László. Művészeti szerkesztő: Pribelszky Pál
Tördelő: Domokos Imre. Hírrovat: Csányi György. Vallató: Pogány György. Gépnyerő: Király Zoltán. Posta: Halász Péter